

Министерство образования и науки РФ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Уральский государственный педагогический университет»
Институт математики, информатики и информационных технологий
Кафедра информационно-коммуникационных технологий в образовании

ПРОГРАММА АВТОМАТИЗАЦИИ ПРОЦЕССА СОСТАВЛЕНИЯ И ФОРМИРОВАНИЯ ТЕХНИЧЕСКОГО ЗАДАНИЯ ДЛЯ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Выпускная квалификационная работа

бакалавра по направлению подготовки

230400 – Информационные системы и технологии

Исполнитель: студент группы ИТ-41

Института математики, информатики и ИТ
Голяка М. А.

Руководитель: к.п.н., доцент кафедры НИТО
Сардак Л.В.

Работа допущена к защите

«12» мая 2016 г.

Зав. кафедрой _____

Екатеринбург – 2016

РЕФЕРАТ

Голяка М. А. ПРОГРАММНЫЙ ПРОДУКТ «ФОРМИРОВАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ» ДЛЯ КАФЕДРЫ, выпускная квалификационная работа: 68 стр., рис. 19, табл. 4, библи. 32 назв., приложений 2.

Ключевые слова: ПРОГРАММНЫЙ ПРОДУКТ, РАЗРАБОТКА VBA ПРИЛОЖЕНИЯ, MS OFFICE, АВТОМАТИЗИРОВАННОЕ ФОРМИРОВАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ, ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ.

Объект разработки – автоматизированная система формирования технического задания.

Цель работы – разработать программу для автоматизации процесса составления и формирования технического задания для выпускных квалификационных работ.

В работе описан процесс проектирования и разработки программного продукта, реализующего автоматизированное составление и формирование технического задания для выпускников учебного заведения. Рассматривается построение его функциональной модели и алгоритма работы.

Разработанная программа реализована с использованием пакета MS Excel и его встроенного языка программирования Visual Basic for Application, шаблон конечного документа технического задания разработан с помощью MS Word.

Программа прошла апробацию в Уральском государственном университете.

Оглавление

Введение.....	4
Глава 1. Автоматизация подготовки текстовой документации по заданной форме на примере технического задания	7
1.1 Анализ программных решений по формированию технического задания.....	7
1.2 Технологии автоматизации подготовки текстовых документов по определенной форме и структуре.....	18
1.3 Техническое задание	22
Глава 2. Реализация «Программы автоматизации процесса составления и формирования технического задания для выпускной квалификационной работы» средствами VBA.....	30
2.1 Моделирование программы «Автоматизация процесса составления и формирования технического задания для выпускной квалификационной работы».....	30
2.2 Разработка «Программы автоматизации процесса составления и формирования технического задания для выпускной квалификационной работы».....	39
2.3 Техническая документация	47
2.4 Апробация результатов работы.....	51
Заключение	52
Список информационных источников.....	54
Приложения	57
Приложение 1	57
Приложение 2	61

Введение

Разработка программного обеспечения – это сложный неоднородный процесс, который должен охватывать множество этапов – от моделирования бизнес-процессов, которые планируется реализовать в виде программного продукта, до процессов практического внедрения и сопровождения конечной системы. Как и многие процессы в ИТ-сфере, разработка программного обеспечения требует наличия определенной структуры, которая соответствует определенному стандарту, т.к. только так может быть достигнута совместимость разрабатываемых продуктов и платформ на которых они функционируют. Структурированность процесса разработки программного обеспечения, в свою очередь, также позволяет обеспечить разграничение проекта по срокам, выделение на этапы определенных ресурсов, а также закрепление ответственных исполнителей на каждом этапе. Подобное описание процесса разработки программного обеспечения, как правило, называется моделью жизненного цикла, описывающей полный цикл разработки.

Большинство проектов по разработке программного обеспечения и информационных систем начинаются с того, что некоторому заказчику, необходима автоматизация некоторых функций, при этом разработчик готов оказать услуги по автоматизации этих функций, путем создания программного продукта. Так начинается технологический процесс разработки программного обеспечения. В ходе выполнения работ должен быть решен ряд задач и проблем, с которыми могут столкнуться разработчики. Большинство проблем в разработке ПО, как показывает практика, связаны с организационными моментами, в частности это проблемы эффективного управления проектом и составление технического задания. Стоит четко понимать, что даже самые хорошие методы и эффективные инструменты теряют свои достоинства, если некорректно составлено техническое задание на проект. По этой причине целесообразно формализовать процесс создания технического задания, для чего возможно внедрение некоторого программного продукта. К тому же, если разработка технических заданий выполняется с некоторой периодичностью, как это

делается при написании выпускных работ студентами ИТ-специальностей, то целесообразно создать некоторую автоматизированную систему.

Необходимо учесть, что требования по содержанию технического задания уже давно установлены в российских ГОСТ и международных ISO стандартах, но, к сожалению, отдельные разработчики не придерживаются данных рекомендаций, предпочитая разработку собственных стандартов. Это приводит к серьезной путанице, при разработке, приемке и дальнейшем сопровождении программных продуктов. Для учебных заведений подобная ситуация недопустима – все выпускники, как будущие специалисты, должны понимать, что при создании технического задания на свои разработки, они должны придерживаться существующих стандартов.

Актуальность настоящего дипломного проекта заключается в том, что в ней проводится аналитический обзор процесса разработки программного обеспечения, по результатам которого производится создание программного продукта для автоматизированного формирования технического задания.

Цель – разработать программу для автоматизации процесса составления и формирования текстового файла в docx формате с техническим заданием для выпускных квалификационных работ инженерных направлений подготовки.

Объект разработки – автоматизация процесса формирования текстового файла технического задания на разработку операционной системы, системы управления базами данных, игровой программы, интегрированного пакета, средства для обработки цифровых данных, языка программирования, сервисной программы.

Основные задачи, решаемые в работе:

- проведение анализа существующих программных решений по автоматизации процесса подготовки технического задания;
- проведение анализа документации, формализующей предметную область;
- разработка программного продукта и документации на него;

- проведение апробации программного продукта.

Глава 1. Автоматизация подготовки текстовой документации по заданной форме на примере технического задания

1.1 Анализ программных решений по формированию технического задания

В 1958 всемирно известный статистик Джон Тьюкей (John Tukey) впервые ввел термин software – программное обеспечение. В 1972 году IEEE выпустил первый номер Transactions on Software Engineering – Труды по Программной Инженерии. Первый целостный взгляд на эту область профессиональной деятельности появился в 1979 году, когда Компьютерное Общество IEEE подготовило стандарт IEEE Std 730 по качеству программного обеспечения. После 7 лет напряженной работы, в 1986 году IEEE выпустило IEEE Std 1002 «Taxonomy of Software Engineering Standards».

Наконец, в 1990 году началось планирование всеобъемлющих международных стандартов, в основу которых легли концепции и взгляды стандарта IEEE Std 1074 и результатов работы образованной в 1987 году совместной комиссии ISO/IEC JTC 1. В 1995 году группа этой комиссии SC7 «Software Engineering» выпустила первую версию международного стандарта ISO/IEC 12207 «Software Lifecycle Processes». Этот стандарт стал первым опытом создания единого общего взгляда на программную инженерию. Соответствующий национальный стандарт России – ГОСТ Р ИСО/МЭК 12207-99 [ГОСТ 12207, 1999] содержит полный аутентичный перевод текста международного стандарта ISO/IEC 12207-95 (1995 года).

В свою очередь, IEEE и ACM, начав совместные работы еще в 1993 году с кодекса этики и профессиональной практики в данной области (ACM/IEEE-CS Code of Ethics and Professional Practice), к 2004 году сформулировали два ключевых описания того, что сегодня мы и называем основами программной инженерии – Software Engineering:

1) Guide to the Software Engineering Body of Knowledge (SWEBOK), IEEE 2004 Version - Руководство к Своду Знаний по Программной Инженерии, в дальнейшем просто «SWEBOK» [21];

2) Software Engineering 2004. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering – Учебный План для Преподавания Программной Инженерии в ВУЗах [12].

Результатом разработки данных стандартов стало появление такой области знаний, как «программная инженерия» или «процесс разработки программного обеспечения».

Область знаний «Процесс программной инженерии» может быть рассмотрена на двух уровнях. Первый уровень содержит техническую и управленческую деятельность на протяжении процессов жизненного цикла программного обеспечения, включающих приобретение, разработку, сопровождение и вывод из эксплуатации программных систем. Второй уровень – «мета-уровень», связанный с определением, реализацией, оценкой, измерением, управлением, изменением и совершенствованием самих процессов жизненного цикла программного обеспечения.

При анализе отдельного проекта разработки программного продукта рекомендуется стандартизация всех текущих процессов, которая получила название стандартного процесса, по сути представляющего собой некоторую базу данных, содержащую следующее сведения:

- информацию, правила использования, документацию и инсталляционные пакеты средств разработки, используемых в проектах компании (систем версионного контроля, средств контроля ошибок, средств программирования – различных IDE, СУБД и т.д.);
- описание практик разработки – проектного менеджмента, правил работы с заказчиком и т.д.;
- шаблоны проектных документов – технических заданий, проектных спецификаций, планов тестирования и т.д. и пр.

Таким образом, можно прийти к выводу о том, что процесс разработки программного обеспечения – это целый комплекс этапов, которые должны выполняться в определенной последовательности, определенной группой лиц, иметь четкие критерии оценки завершенности и полноты реализации. Это говорит о том, что данные сведения необходимо каким-то образом систематизировать, стандартизировать и дать четкий набор определений. Чтобы систематизировать процесс разработки программного продукта рассматривают его жизненный цикл¹. В жизненном цикле программного продукта выделяют следующие основные этапы:

- анализ требований;
- проектирование;
- кодирование (программирование);
- тестирование и отладка;
- эксплуатация и сопровождение.

Процессы жизненного цикла программных средств также описываются на уровне международных стандартов, наиболее полно их отражает стандарт ISO 12207:1995.[22]

Однако этот стандарт не предлагает конкретной модели жизненного цикла и методов разработки, его рекомендации являются общими для любых моделей жизненного цикла.

1) Водопадная модель жизненного цикла

Водопадная модель жизненного цикла (англ. waterfall model) была предложена Уинстоном Ройсом в 70 гг. 20 века. В рамках данной модели, предполагается выполнение всех этапов проекта строго заданном порядке, причем этот порядок не допускает изменений. Переход к следующему этапу может быть осуществлен только после завершения работ в рамках текущего этапа [17]. Схематически данная модель представлена на рисунке 1.

¹ Жизненный цикл (ЖЦ) программного обеспечения — период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации.



Рисунок 1. Водопадная модель ЖЦ ПО

В соответствии с представленным рисунком можно видеть, что переход к проектированию может быть осуществлен только после того как выработаны все требования и проведен их анализ. Каждая стадия завершается выпуском полного комплекта необходимой и исчерпывающей документации, которой будет достаточно, чтобы работа над проектом могла быть продолжена другими разработчиками. Пропуск одного из этапов, при использовании данной модели ЖЦ ПО недопустим.

Преимущества:

- полная и исчерпывающая документация, формируется на каждом этапе разработки;
- простота определения сроков выполнения проекта и его стоимости;
- возможность продолжения работы над проектом другой командой разработчиков.

Недостатки:

Наибольший недостаток водопадной модели заключается в том, что каждый последующий этап может быть выполнен только после корректного

завершения текущего этапа. При этом обнаруженные ошибки ранних этапов могут быть устранены только в случае возврата к первоначальному этапу. Это приводит к увеличению сроков реализации проекта, а также к дополнительным финансовым затратам.

Таким образом, водопадная модель для крупных проектов мало реалистична и может быть эффективно использована только для создания небольших систем.

2) Итерационная модель

Альтернативой водопадной последовательной модели является модель итеративной разработки, которую также называют эволюционной моделью. Итерационная модель предполагает разбиение ЖЦ ПО на последовательность итераций, каждая из которых представляет отдельный проект, в который входят все предыдущие итерации. Цель выполнения каждой итерации заключается в получении работающей версии системы, выполненной с учетом предыдущих и текущей итерации. Если при получении работающей версии системы обнаруживается какая-либо ошибка, то откат возможен как на самое начало, так и на любой предыдущий шаг, что существенно экономит время на поиск и устранение ошибок [19]. Схематически данная модель представлена на рисунке 2.

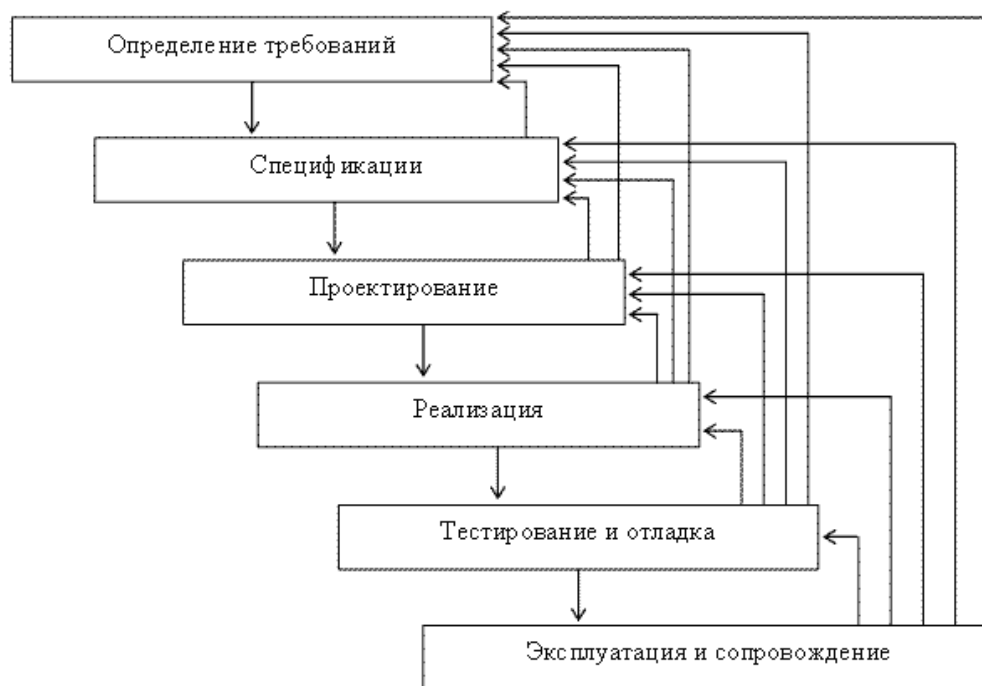


Рисунок 2. Итерационная модель ЖЦ

Основной недостаток итерационной модели заключается в том, что целостное понимание возможностей и ограничений разрабатываемого проекта отсутствует долгое время, т.к. любые проблемы приводят к откату, даже если проблемы связаны с ограничениями, заданными в рамках определения требований. Еще один недостаток, которая присуща проектам, реализуемым по данной модели – добросовестность разработчиков, которая является следствием того, что разработчик не может быть уверен в качестве принятых решений, потому что всегда есть возможность «откатиться и сделать лучше».

3) Спиральная модель

Предложена данная модель в середине 80-х гг. 20 века Барри Боэмом. Суть данной модели заключается в том, что процесс разработки программного продукта включает в себя как проектирование, так и поэтапное прототипирование с целью сочетания преимуществ восходящей и нисходящей концепции, делающая упор на начальные этапы жизненного цикла: анализ и проектирование [16]. Схематически данная модель представлена на рисунке 3.

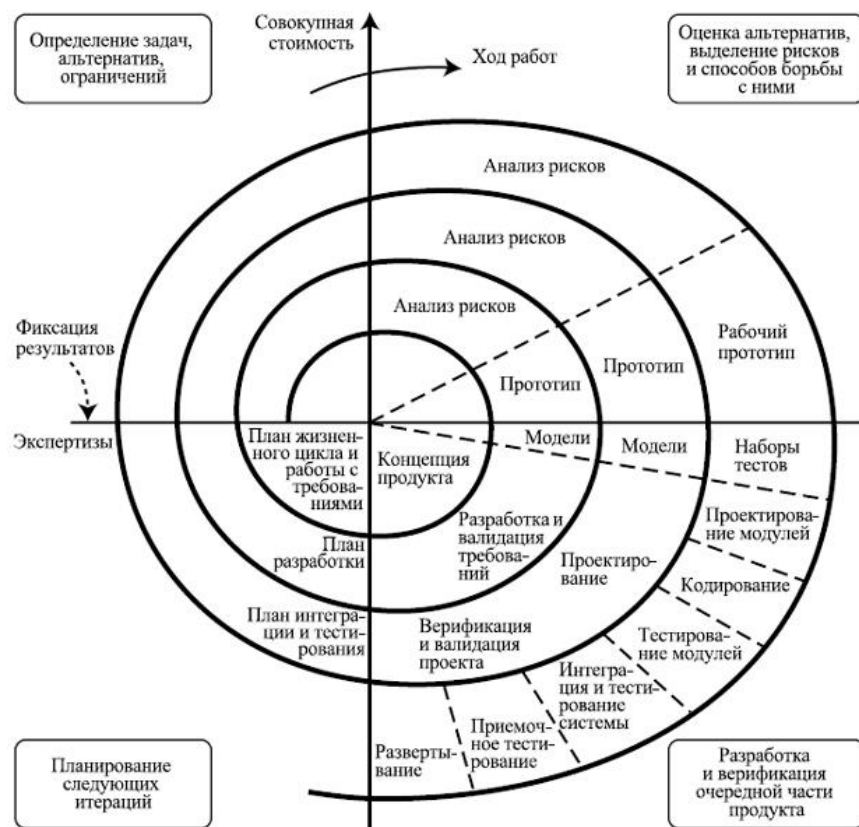


Рисунок 3. Спиральная модель ЖЦ ПО

Каждая итерация соответствует созданию фрагмента или версии ПО, на ней уточняются цели и характеристики проекта, оценивается качество полученных результатов и планируются работы следующей итерации. Каждая выполняемая итерация позволяет оценить:

- риски превышения сроков и стоимости проекта;
- наличие необходимости выполнения еще одной итерации;
- степень полноты и точности понимания требований к системе;
- целесообразность прекращения проекта.

Как в случае с водопадной и итерационной моделями, спиральная модель не лишена недостатков, к основному из них можно отнести – отсутствие четкой регламентации стадий разработки.

К числу преимуществ спиральной модели можно отнести:

- Быстрое получение результата;
- Повышение конкурентоспособности;
- Изменение требований не вызывает серьезных проблем.

Независимо от того, какая модель ЖЦ выбирается под конкретный проект, проектирование программного продукта всегда сопряжено с процессом подготовки и согласования технического задания.

Техническое задание (ТЗ) — исходный документ на проектирование технического объекта (изделия). ТЗ устанавливает основное назначение разрабатываемого объекта, его технические характеристики, показатели качества и технико-экономические требования, предписание по выполнению необходимых стадий создания документации (конструкторской, технологической, программной и т. д.) и её состав, а также специальные требования [1].

Техническое задание является документом, который позволяет как разработчику, так и заказчику представить конечный продукт и впоследствии выполнить проверку на соответствие предъявленным требованиям.

Руководствующими стандартами при написании технического задания являются ГОСТ 34.602.89 «Техническое задание на создание автоматизированной системы» [2] и ГОСТ 19.201-78 «Техническое задание. Требования к содержанию и оформлению» [3]. Первый стандарт предназначен для разработчиков автоматизированных систем, второй для программных средств.

При проектировании программного обеспечения, часто возникают проблемы с подготовкой технического задания, в число которых входят не типизированность и не формализованность. Для решения этих проблем необходимо подготовить общую типизированную, частично формализованную форму технического задания. Такая форма должна быть универсальной — подходить для разработки различных программных продуктов и при этом соответствовать установленным требованиям ГОСТа.

Как правило, на создание проектной документации затрачивается много времени и усилий разработчика, в связи с чем на данный этап ЖЦ программного продукта выделяется больше времени, чем нужно. В рамках срочных проектов данный этап может быть вообще упущен, в пользу распределения ресурсов времени на другие, более важные задачи. При таком подходе на последующих этапах ЖЦ программного продукта возникает множество

проблем – перерасход имеющихся ресурсов, проблемы в кодировании, непонимание между разработчиком и заказчиком и т.д.

В качестве выхода из сложившейся ситуации прибегают к использованию специальных программных продуктов, которые позволяют автоматизировать процесс создания технической документации к программным продуктам. На сегодняшний день, на рынке представлены различные реализации подобных систем, одни позволяют формировать документацию в соответствии с некоторыми предустановленным шаблонам, другие формируют документацию путем извлечения исходной информации из репозитариев и артефактов проекта [5]. Подобные системы обладают избыточной функциональностью, имеют техническую поддержку, могут охватывать полный цикл проекта создания программного продукта, бесплатные версии не поддерживают отечественных стандартов и т.д. Стоимость «коробочных» продуктов ограничивает их распространенность, особенно для учебных целей или в небольших некоммерческих проектах.

В некоторых случаях прибегают к самостоятельной разработке программных продуктов, используемых для создания технической документации. Такой подход обеспечивает существенно снизить затраты на внедрение программных инструментов для создания технической документации. Кроме того, самостоятельная разработка позволит учесть соблюдение внутренних требований и стандартов организации, создающей такую систему. Еще одно преимущество такого подхода заключается в том, что при самостоятельной разработке на практике закрепляются знания о методологии создания программной документации, в частности технического задания на разработку программного продукта.

В рамках данного дипломного проекта ставится задача автоматизации создания технического задания на разработку программного продукта для выпускников учебного заведения, темой выпускной работы которых является разработка программ. Как уже было сказано ранее, подобная задача может

быть решена несколькими способами – с помощью использования «коробочного» продукта или с помощью самостоятельной разработки. Для начала рассмотрим несколько программных продуктов, с помощью которых может быть создано ТЗ в автоматическом режиме, к таким системам относятся:

- TZ-master;
- IBM Rational SoDA;
- Requirements Miner for AllFusion Modeling Suite;
- IBM Rational Telelogic DOORS;
- Sybase Power Designer;
- Business Studio 3;
- Author IT.

Существуют также и другие системы, которые в чем-то могут превосходить представленные, но рассматривать их не имеет смысла, т.к. указанный перечень программ является наиболее оптимальным для оценки автоматизированного создания систем технической документации на разработку программных продуктов [8].

В таблице 1 проведен сравнительный анализ указанных программных продуктов, путем сравнения основного набора требований, которым должен соответствовать программный продукт, удовлетворяющий задачам дипломной работы.

Таблица 1.
Сравнительный анализ программных продуктов

Критерии	TZ-master	IBM Rational SoDA	Requirements Miner for AllFusion Modeling Suite	Sybase Power Designer	IBM Rational Telelogic DOORS	Business Studio 3	Author IT
Соблюдение ГОСТов РФ:							
ГОСТ 19.004-80	-	-	-	-	-	-	+
ГОСТ 2.114-95	-	-	-	-	-	-	+
ГОСТ 19.201-78	-	-	-	-	-	-	+
ГОСТ 34.602-89	-	-	-	-	-	-	+

Генерация конечного документа/отчета в Word	+	+	+	+	+	+	+
Возможность работы с графическими данными	+	+	+	+	+	+	+
Поддержка русифицированных шаблонов и отчетов	+	+	+	-	-	+	+
Цена производителя на 01.11.2011	Бесплатно	11800 рублей	90749 рублей	130000 рублей	43400 рублей	9800 рублей	25600 рублей
Интерфейс	Русский	Английский	Русский	Английский	Английский	Русский	Английский
Поддерживаемые ОС	Windows 2000; XP; Vista; Seven.	Windows XP; Vista; Seven; Linux.	Windows XP; Vista; Seven; Linux.	Windows XP; Vista; Server 2003.	Windows; Linux.	Windows; Linux.	Windows; Linux.
Предварительный просмотр	+	+	+	+	+	+	+

Указанные программные продукты позволяют выполнять возложенные на них функции по формированию технического задания на разработку программного продукта, но при этом не отвечают требованиям, которым должны соответствовать для наиболее полного достижения задач дипломного проекта. Причем основной недостаток перечисленных программ – это несоответствие российским ГОСТ и высокая стоимость [16].

Таким образом, наиболее оптимальным вариантом решения поставленной задачи является создание собственного программного продукта, который

позволит автоматизировать процесс составления технического задания для выпускной квалификационной работы студентов ИМИиИТ. Разработанная программа позволит учесть внутренние стандарты учебного заведения, а также позволят создать бесплатный, не перегруженный ненужными функциями программный продукт.

1.2 Технологии автоматизации подготовки текстовых документов по определенной форме и структуре

Под методами разработки программного продукта понимаются те действия, которые следуют от определения и формализации решения о необходимости создания программы до момента пока программа не будет введена в эксплуатацию. На сегодняшний день существуют такие методы разработки программ:

- разработка нового программного продукта;
- покупка готовой программы и ее доработка под заданные требования.

Для понимания отличия между указанными вариантами рассмотрим каждый из методов детально.

При разработке нового программного продукта увеличиваются сроки, необходимые для полноценного создания программы. Но при таком подходе конечный программный продукт будет наиболее полно соответствовать всем требованиям, которым должна соответствовать разрабатываемая программа. Такой метод разработки наилучшим образом подойдет в учебных целях, когда несколько важен результат работы, как важен сам процесс разработки. В рамках дипломного проектирования выбирается именно данный метод.

Рассмотрим также и такой метод, при котором покупается готовая программа с открытым (частично открытым) исходным кодом и самостоятельно дорабатывается. Данный подход выгоден для автоматизации более сложных процессов, которые чаще протекают в бизнес-среде. Ранее было установлено,

что требуемый программный продукт не будет закупаться в готовом виде, а будет полностью создаваться студентом-выпускником в рамках дипломного проектирования [9]. В связи с этим, в рамках данного дипломного проекта ставится задача по разработке программного продукта, который позволит разработать программный продукт, необходимый для автоматизации процесса составления и формирования технического задания на разработку программного продукта, в соответствии со стандартами и требованиями, принятыми в ИМИ и ИТ. Создание подобной программы может быть выполнено с использованием различных средств, например:

- Среда разработки Borland Delphi;
- Среда разработки Borland C++ Builder
- Среда разработки MS Visual Studio;
- Пакет MS Office 2013.

Кратко рассмотрим перечисленные программные продукты.

Borland Delphi, последней версией которой на сегодняшний день является Borland Delphi 8.0, – новая редакция кросс-платформенной среды для быстрой разработки приложений Delphi. Основное отличие этого продукта от предыдущих версий - его ориентация на технологическую платформу Microsoft .NET. Borland Delphi 8 дает возможность разработчикам, пользующимся средой Delphi, создавать программные средства на основе .NET Framework, однако сохранена и возможность создания приложений на основе WinAPI [8].

Основное направление, на которое делается упор при разработке приложений в Delphi – работа с базами данных. Естественно, такая строгая направленность средства, появившаяся с первыми версиями этой интегрированной среды разработки постепенно уходит в прошлое, но, тем не менее, Delphi по-прежнему остается одним из самых популярных средств для разработки баз данных.

Кроме того, Delphi относится к средствам так называемой быстрой разработки приложений (RAD – rapidly application development), то есть акцент

делается на как можно более оперативную реализацию готового проекта. Для этого применяются различные визуальные средства проектирования, задача которых предоставить возможность создания программы из готовых компонентов при минимальном количестве кода.

Естественно, что при использовании готовых компонентов в программу неизбежно попадают неиспользуемые блоки, наличие которых плохо сказывается как на производительности программы, так и на ее общей ресурсоемкости [18].

Borland C++ Builder стал одним из самых популярных на сегодняшний день инструментов для создания как настольных, так и корпоративных информационных систем благодаря уникальному сочетанию удобства разработки пользовательских интерфейсов, компонентной архитектуры, однотипности доступа к разнообразным базам данных [9]. Это средство быстрой разработки приложений, позволяющее создавать приложения на языке C++, используя при этом среду разработки и библиотеку компонентов Delphi [10]. Отличие этих двух сред обусловлено в основном только разным синтаксисом языков: в Delphi используется Object Pascal, в C++ Builder – язык C++.

С помощью C++ Builder также создаются приложения, состоящие из набора компонентов. Кроме того, для того, чтобы такие приложения работали необходимо наличие в системе специальной библиотеки процедур (впрочем, такая библиотека может быть интегрирована с самой программой), что также не сказывается благоприятно на производительности и ресурсоемкости [23].

Microsoft Visual Studio .NET – обеспечивает каркас общей среды, на которой базируются несколько языков. Она позволяет создавать приложения и Web-службы, которые поддерживают множество функций и легко масштабируются. Visual Studio поддерживает Visual Basic .NET (VB.NET), C# (C-sharp), C++ и J# (J-sharp) [11].

Интегрированная среда Microsoft Visual Studio позволяет создавать высокопроизводительные системы, включает в себя специальный оптимизатор

кода. Основная задача Microsoft Visual Studio – создание приложений для Windows. Именно с помощью этой интегрированной среды разработки было создано большинство продуктов фирмы Microsoft и других известных производителей.

В качестве инструмента для создания системы автоматического создания технического задания для студентов, выбирается программный продукт компании Microsoft – MS Excel 2013 и его инструмент VBA. Разрабатываемая программа должна представлять собой программный продукт, обеспечивающий интерфейс взаимодействия с пользователем. Именно поэтому выбирается программный продукт MS Excel, имеющий в своем составе средства разработки на встроенном языке программирования VBA [29].

Microsoft Excel представляет собой программу для работы с электронными таблицами, он обладает множеством инструментов, которые возможно использовать при автоматизации процесса создания технического задания. Ценной возможностью Excel является возможность писать код на основе Visual Basic для приложений (VBA). Этот код пишется с использованием отдельного от таблиц редактора. Управление электронной таблицей осуществляется посредством объектно-ориентированной модели кода и данных. С помощью этого кода данные входных таблиц будут мгновенно обрабатываться и отображаться в таблицах и диаграммах (графиках). Таблица становится интерфейсом кода, позволяя легко работать изменять его и управлять расчетами. Это означает, что если имеется электронная таблица, содержащая в своих строках определенные данные, то проводя небольшие изменения, ее можно использовать и для других случаев, т.е. сделать из нее шаблон.

Кроме того, разработка скриптов VBA осуществляется достаточно просто и не требует изучения тонкостей программирования, которые потребуются для использования таких сред как MS Visual Studio или Borland Delphi. Для использования VBA не требуется покупка дополнительного программного обеспечения, т.к. он поставляется в комплекте MS Office 2013 и достаточно

распространен на компьютерах, установленных в аудиториях учебного заведения.

Таким образом, MS Excel VBA идеально подойдет для решения задачи разработки системы автоматизированного формирования технического задания на программные продукты, разрабатываемые выпускниками.

1.3 Техническое задание

на разработку «Программы автоматизации процесса составления и формирования технического задания для выпускной квалификационной работы»

Составлен на основе ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы»

1. Общие сведения.

1.1. Название организации-заказчика.

Кафедра ИКТО, УрГПУ

1.2. Назначение продукта.

Программный продукт позволяет формировать технические задания на разработку программ в автоматическом режиме, основываясь на данных вводимых пользователем.

1.3. Плановые сроки начала и окончания работ.

Таблица 2.
Календарный план

№ п/п	Этапы выполнения	Сроки выполнения этапа	
		Начало периода	Конец пери- ода
1.	Разработка требований	01.01.2016	05.01.2016
2.	Проектирование	06.01.2016	20.01.2016
3.	Реализация	01.02.2016	01.04.2016
4.	Тестирование	02.04.2016	04.04.2016

№ п/п	Этапы выполнения	Сроки выполнения этапа	
		Начало периода	Конец пери- ода
5.	Ввод в действие	-	-

2. Характеристика области применения продукта.

2.1. Процессы и структуры, в которых предполагается использование продукта разработки.

Использование продукта предполагается в образовательных целях. Студенты, при выполнении разработки программного продукта смогут самостоятельно составлять технического задание, на разрабатываемое ими программное обеспечение, используя результаты разработки в качестве шаблона. Основным процесс, в котором будет задействован разрабатываемый программный продукт – формирование технической документации на программное обеспечение.

2.2. Характеристика персонала (количество, квалификация, степень готовности).

Таблица 3.
Характеристика персонала

	Квалификация	Степень готовности
Рабочие	-	-
Руководители	Преподаватель с научной степенью, ответственный за промежуточную проверку функционала программного продукта и окончательную оценку законченного программного продукта.	Базовые навыки работы с операционными системами Microsoft, офисным программным обеспечением Microsoft Office 2003, 2007 и т.д.
Специалисты	Студент-выпускник, непосредственно ответственный за разработку программы	Базовые навыки работы с операционными системами Microsoft, офисным программным обеспечением Microsoft Office 2003, 2007 и т.д.
Технические исполнители	-	-

3. Требования к продукту разработки.

3.1. Требования к продукту в целом.

Программный продукт должен соответствовать следующим требованиям:

- ГОСТ 19.004-80;
- ГОСТ 2.114-95;
- ГОСТ 19.201-78;
- ГОСТ 34.602-89;
- возможность генерации конечного документа/отчета в MS Word;
- возможность работы с графическими данными;
- поддержка русифицированных шаблонов и отчетов;
- бесплатное распространение;
- интуитивно понятный интерфейс;
- поддержка различных ОС (Linux, Unix, Windows и т.д.).

3.2. Аппаратные требования.

Таблица 4.
Аппаратные требования

	Минимальные	Рекомендуемые
Процессор	Процессор с частотой 500 МГц	1 ГГц и выше, 32- или 64-разрядный процессор
Оперативная память	256 МБ ОЗУ	1 ГБ ОЗУ (32-разрядный выпуск); 2 ГБ ОЗУ (64-разрядный выпуск)
Видеоадаптер и монитор	Графический процессор, совместимый с технологией Microsoft DirectX 9.0c, с 64 МБ видеопамятью.	Видеокарта с поддержкой DirectX10 и экран с разрешением 1024x576 пикселей или выше
Свободное место на HDD	1,5 ГБ	3,0 ГБ

3.3. Указание системного программного обеспечения.

Операционная система семейства MS Windows (XP, 2003, Vista, 7 и более поздние).

3.4. Указание программного обеспечения, используемого для реализации.

Установленный офисный пакет Microsoft Office 2007 и более с MS Word и MS Excel с функционирующим редактором VBA и наличием необходимых библиотек:

- Visual Basic For Applications;
- Microsoft Excel X.X Object Library (где X.X – версия библиотеки в зависимости от версии MS Office);
- OLE Automation
- Microsoft Office X.X Object Library (где X.X – версия библиотеки в зависимости от версии MS Office);
- Microsoft Forms 2.0 Object Library;
- Microsoft Word X.X Object Library (где X.X – версия библиотеки в зависимости от версии MS Office).

3.5. Для сетевых систем – особенности реализации серверной и клиентской частей.

Программный продукт не является распределенной системой, при распределении доступа к нему по сети, пока первый пользователь, получивший доступ к файлу с программой может его редактировать, остальные пользователи могут работать только в режиме просмотра.

3.6. Форматы входных и выходных данных.

В MS Word в качестве входных форматов файлов используются .jpg и xlsx.

В MS Excel в качестве входных данных используется текстовая информация.

В качестве файла для выходных данных, используются docx-файлы, частности в данном примере используется файл «pattern.docx».

3.7. Источники данных и порядок их ввода в систему (программу), порядок вывода, хранения.

В качестве исходных данных для системы выступает файл-шаблон «start.xlsm». Данный файл имеет заранее определенную структуру и позволяет заполнять его пользовательскими сведениями. Ввод данных осуществляется пользователем с клавиатуры. Введенные в шаблонный файл данные не записываются и при последующем запуске не отображаются, но при формировании конечного документа – технического задания, эти данные заносятся в бланк технического задания.

3.8. Порядок взаимодействия с другими системами, возможности обмена информацией.

Разрабатываемый программный продукт взаимодействует с приложениями, входящими в состав MS Office различных версий, с помощью которых обрабатывается пользователем.

Разрабатываемый программный продукт взаимодействует с программами архиваторами (при необходимости), для уменьшения занимаемого им места на носителе информации.

Формируемые выходные файлы могут быть обработаны или занесены в прочие программные продукты, имеющие возможность импорта doc-файлов.

3.9. Меры защиты информации:

Меры по защите программного кода не используются, программный продукт является свободно распространяемым приложением.

В MS Excel присутствует защита ячеек, осуществленная стандартными средствами Microsoft Office.

4. Требования к пользовательскому интерфейсу.

4.1. Общая характеристика пользовательского интерфейса.

Интерфейс должен обеспечивать удобную навигацию в диалоге с пользователем, который хорошо знает свою предметную область и не является специалистом в области автоматизации.

Наличие контекстно-зависимой помощи.

4.2. Размещение информации на экране, дизайн экрана.

Настраиваемость графических элементов интерфейса, в том числе цветового оформления, в пределах возможностей MS Excel.

Информационные блоки должны быть размещены независимо друг от друга и пересекаться.

Не должно быть скрытых элементов на экране (исполняемых, но не доступных для управления пользователем).

Техническое задание

на разработку

Название системы:

1. Общие сведения

1) Название организации-заказчика:

2) Назначение системы:

3) Плановые сроки начала и окончания работ

Этап выполнения	Начало периода	Конец периода
1. Разработка требований	24.01.1900	25.01.1900
2. Проектирование	26.01.1900	27.01.1900
3. Реализация	08.01.1900	09.01.1900
4. Тестирование	10.01.1900	11.01.1900
5. Ввод в действие	12.01.1900	13.01.1900

2. Характеристика области применения продукта

1) Процессы и структуры, в которых предполагается использование продукта разработки

Использование продукта предполагается в

2) Характеристика персонала

	Квалификация	Степень готовности
<input type="checkbox"/> Рабочие		
<input type="checkbox"/> Руководители		
<input type="checkbox"/> Специалисты		
<input type="checkbox"/> Технические исполнители		

Рисунок 4. Макет интерфейса

1.1 Техническое задание

на разработку &tem1

«&tem3».

Составлен на основе ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы»

1. Общие сведения.

1.1. Название организации-заказчика.

&tem4

1.2. Назначение ИС

&tem5

1.3. Плановые сроки начала и окончания работ.

Таблица 1.
Календарный план

№ п/п	Этапы выполнения	Сроки выполнения этапа	
		Начало периода	Конец периода
1.	Разработка требований	&Data1	&Data2
2.	Разработка требований	&Data3	&Data4
3.	Реализация	&Data5	&Data6
4.	Тестирование	&Data7	&Data8
5.	Ввод в действие	&Data9	&Dataa1

2. Характеристика области применения продукта.

2.1. Процессы и структуры, в которых предполагается использование продукта разработки

Использование продукта предполагается в &tem6.

2.2. Характеристика персонала (количество, квалификация, степень готовности)

Таблица 2.
Характеристика персонала

Рисунок 5. Внешний вид шаблона Word

4.3. Особенности ввода информации пользователем, предоставление вы- ходных

Разрабатываемая система должна производить блокировку некорректных действий пользователя при работе с системой.

Пользователь должен корректно вводить исходные данные.

Пользователь должен иметь возможность предварительного просмотра конечного документа, который сформируется после окончания ввода.

5. Требования к документированию.

5.1. Перечень сопроводительной документации.

В ходе создания программного продукта должен быть подготовлен и передан комплект документации в составе:

- техническое задание на разработку программного продукта;
- сопроводительная документация на разработанный программный продукт;
- руководство пользователя;

- руководство администратора.

Состав и содержание комплекта документации на разрабатываемый программный продукт может быть уточнен на стадии проектирования и пересмотрен при сдаче-приемке программного продукта.

5.2. Требования к содержанию отдельных документов.

Документы должны быть представлены на бумажном виде (оригинал) и на переносимом носителе (копия). Исходные тексты программ – только на переносимом носителе (оригинал).

Все документы должны быть оформлены на русском языке. Состав документов на общее программное обеспечение, поставляемое в составе программного продукта, должен соответствовать установленному комплекту поставки.

Подготовленные документы должны удовлетворять требованиям государственных стандартов и рекомендаций по оформлению, содержанию, форматированию, использованию терминов, определений и надписей, обозначений программ и программных документов.

6. Порядок сдачи-приемки продукта.

В соответствии с графиком подготовки ВКР.

Глава 2. Реализация «Программы автоматизации процесса составления и формирования технического задания для выпускной квалификационной работы» средствами VBA

2.1 Моделирование программы «Автоматизация процесса составления и формирования технического задания для выпускной квалификационной работы»

Проектирование программного продукта начинается с разработки его модели. Ввиду того, что конечная программа должна автоматизировать некоторый бизнес-процесс, то целесообразно применять для моделирования его работы диаграммы IDEF0, используемые для формализации бизнес-процессов [26].

Для создания IDEF0-модели работы программы автоматизации процесса составления и формирования технического задания используется специализированное средство моделирования – BrWin, которое позволяет построить диаграмму описания бизнес-процесса, с использованием нотификации IDEF0 и строить ее декомпозиции.

Перед проектированием необходимо выявить основные свойства бизнес-процессов. Любой бизнес-процесс обладает следующими характеристиками:

- результативность бизнес-процесса — соответствие результатов бизнес-процесса нуждам и ожиданиям клиентов.
- эффективность бизнес-процесса — способность бизнес-процесса обеспечивать заданную результативность при минимальном расходовании ресурсов и отсутствии непроизводительных трат и потерь.
- определенность бизнес-процесса — степень, с которой реальный бизнес-процесс соответствует своему документальному описанию.
- повторяемость бизнес-процесса — способность бизнес-процесса создавать выходные потоки (результаты) с одинаковыми характеристиками при повторных его реализациях.

- адаптируемость бизнес-процесса — характеристика гибкости бизнес-процесса, т. е. способности соответствовать будущим изменениям потребительских требований или удовлетворять особые текущие потребности некоторых из них.

Эффективность и адаптируемость бизнес-процесса обработки является низкой ввиду высокой нормы расходования ресурсов и практического отсутствия гибкости. Результативность, определенность и повторяемость в свою очередь достаточно высоки.

Таким образом, проектирование программного продукта для автоматизации составления и формирования технического задания, должно быть осуществлено таким образом, чтобы конечная система обладала совместимостью с существующей информационно-вычислительной системой и обладала возможностью интеграции в нее [31].

Со стороны аппаратной части необходимо минимизировать вносимое в существующую систему оборудование. С программной стороны – необходимо автоматизировать взаимодействие между отдельными модулями системы.

Рассмотрим бизнес-процесс формированию технического задания. Диаграмма данного бизнес-процесса представлена на рисунке 6.

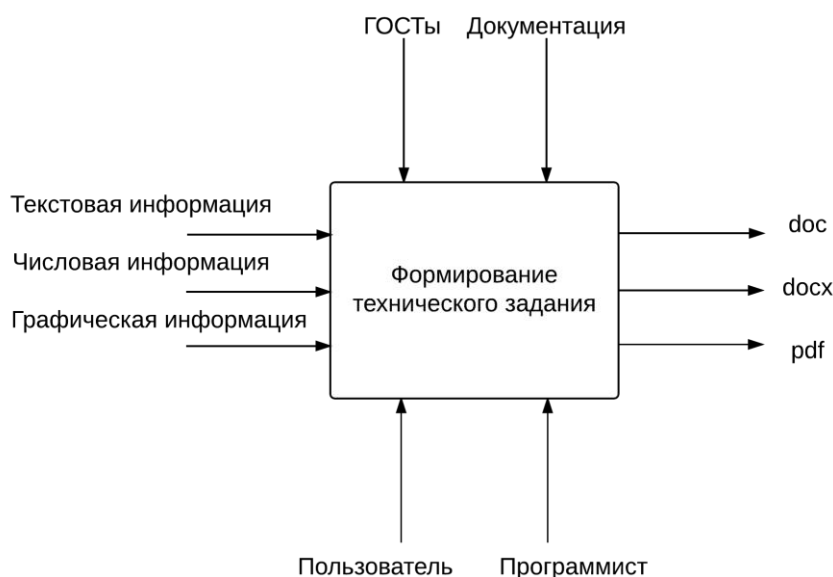


Рисунок 6. Контекстная диаграмма
IDEF0 – «Формирование технического задания»

В данном случае деятельность по формированию технического задания представляется в виде «черного ящика», а взаимодействие системы с окружающей средой описывается в терминах входа, выхода, механизмов и управления.

После описания контекстной диаграммы проводится функциональная декомпозиция - система разбивается на подсистемы и каждая подсистема описывается отдельно (диаграммы декомпозиции). Затем каждая подсистема разбивается на более мелкие и так далее до достижения нужной степени подробности. В рамках данной работы рассмотрим только декомпозицию первого уровня, которая представлена на рисунке 7.

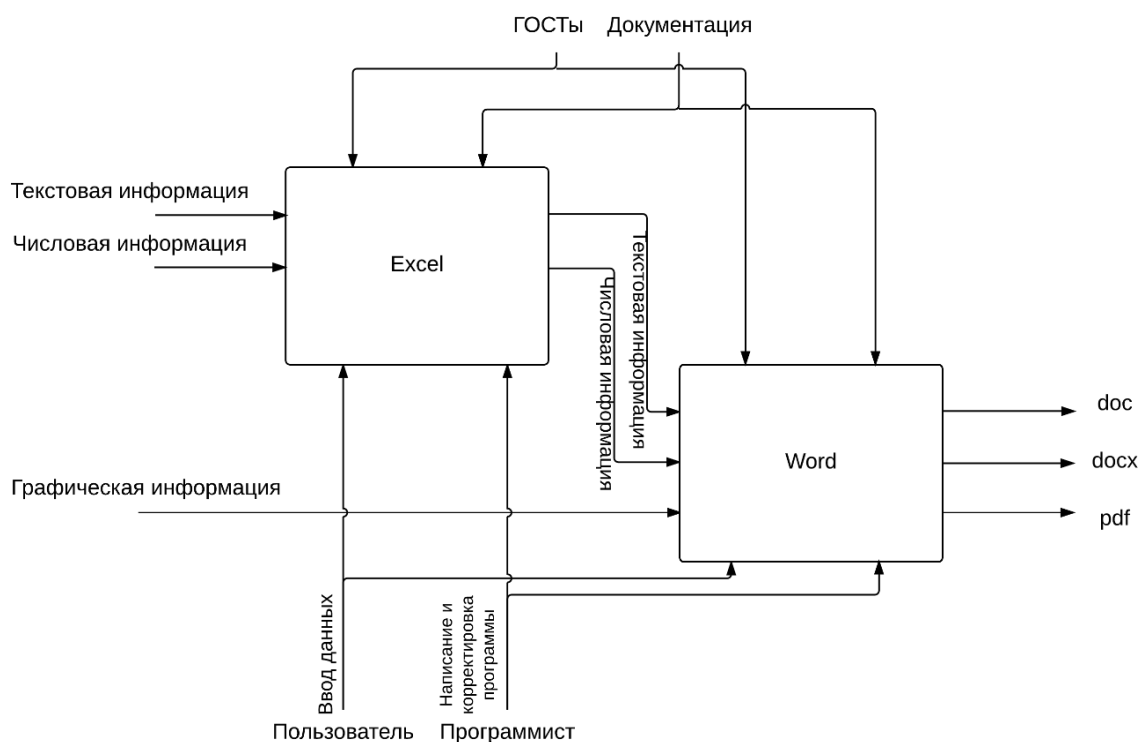


Рисунок 7. Декомпозиция контекстной диаграммы «Формирование технического задания»

Таким образом, из представленной диаграммы декомпозиции можно видеть, что процесс формирования технического задания осуществляется следующим образом: пользователь заполняет данные в программе MS Excel, руководствуясь требованиями, представленными в нормативной документации, затем

введенные данные передаются в программу MS Excel. То, каким образом производится преобразование данных, пользователь не должен видеть, т.к. все это должно быть скрыто от него в программном коде [12].

Перед непосредственной разработкой программного продукта рассмотрим алгоритм формирования технического задания в виде блок-схемы. Необходимость визуализации алгоритма выполнения программы поможет для дальнейшей разработки программного кода. Блок-схема работы программы для автоматизации формирования и создания технического задания на разработку программного продукта представлена на рисунке 8.



Рисунок 8. Блок-схема алгоритма работы программы автоматизированного формирования технического задания

На основании блок-схемы рассматриваемого алгоритма можно построить модель жизненного цикла программного продукта для автоматизированного формирования и создания технического задания. При построении модели жизненного цикла программного продукта, необходимо учитывать требования группы стандартов ГОСТ 34, а также те аспекты, которые были рассмотрены в предыдущей главе дипломного проектирования. Согласно этим требованиям программный продукт будет содержать следующие этапы жизненного цикла:

- исследование предметной области с последующим формированием функциональной и информационной моделей объекта, для которого предназначен программный продукт;
- проектирование программного продукта, заключающееся в разработке проектных решений, удовлетворяющих всем требованиям ТЗ;
- разработку программного продукта (в том числе программирование и тестирование прикладных программ на основании проектных спецификаций подсистем, выделенных на стадии проектирования);
- тестирование программного продукта и выявление сбоев с последующим их устранением;
- эксплуатация программного продукта и его сопровождение;
- развитие программного продукта.

Рассмотрим каждый из перечисленных этапов более подробно.

1-й этап Исследование предметной области

На данном этапе разработки проекта осуществляется анализ предметной области, который был выполнен ранее, следовательно, на данный момент работы данного этапа считаются выполненными.

Основная цель данного этапа – собрать подробную информацию по формированию технического задания на разработку программных продуктов, осуществляемых в учебных целях;

Ключевые участники данного этапа – студент, ответственный за разработку проекта, научный руководитель, выступающий в качестве лица, ответственного за качество исполнения работ и утверждающий правильность и актуальность полученной информации.

Требования к входной информации - информация должна быть полной и достоверной.

Полученные результаты – получены все необходимые сведения для начала работ по проектированию.

2-й этап Проектирование программного продукта

Проектирование (техническое и логическое проектирование). В соответствии с требованиями формируются состав автоматизируемых функций (функциональная архитектура) и состав обеспечивающих подсистем (системная архитектура), проводится оформление технического проекта. На данном этапе выбираются не только автоматизируемые функции, но и производится выбор программных средств, за счет которых будет производиться автоматизация. Таким образом, этап проектирования предполагает:

- проектирование функциональной архитектуры программного продукта, которая отражает структуру выполняемых функций;
- проектирование системной архитектуры программного продукта (состав обеспечивающих подсистем);
- реализацию проекта;
- составление списка необходимой документации.

Формирование функциональной архитектуры, которая представляет собой совокупность функциональных подсистем и связей между ними, является наиболее ответственным и важным этапом с точки зрения качества всей последующей разработки программного продукта.

Построение системной архитектуры на основе функциональной предполагает определение элементов и модулей информационного, технического, программного обеспечения и других обеспечивающих подсистем, связей по

информации и управлению между выделенными элементами и разработку технологии обработки информации.

Основная цель данного этапа – разработать проект программы автоматизирующей составление и формирование технического задания на разработку программного продукта.

Ключевые участники этапа - студент, ответственный за разработку проекта, научный руководитель, который проверяет правильность разработанного проекта и возможность его реализации, а также представитель кафедры, который утверждает окончательный проект.

Требования к входной информации - информация должна быть полной и достаточной для создания полноценного проекта разрабатываемого программного продукта.

Полученные результаты – получены все необходимые сведения для начала работ по проектированию.

Мероприятия, описанные на данном этапе, представляют собой процесс составления технического задания, который был рассмотрен в предыдущей главе дипломного проекта. Перечень необходимой документации также был определен ранее.

3-й этап Реализация и внедрение программного продукта

Наиболее важный этап жизненного цикла реализуемого программного продукта – это конечная реализация программы. Данный этап включает в себя следующие мероприятия:

- 1) Написание программного кода, в соответствии с установленными требованиями.

- 2) Подготовка условий для внедрения системы, которая заключается в проверке функционала рабочих мест пользователей.

При подготовке рабочих мест пользователей, необходимо удостовериться, что на них установлен пакет MS Office, имеется доступ к локальной сети и достаточное количество дискового пространства для запуска программы и временного хранения данных.

3) Тестовый запуск программного продукта и занесение в него тестовых данных.

4) Составление полного описания реализованной системы.

5) Разработка инструкции пользователя по работе с системой.

6) Обучение пользователей.

Ключевые участники этапа:

- студент, ответственный за разработку проекта, основная обязанность которого – выполнение всех необходимых мероприятий в рамках реализации проекта;

- научный руководитель, проверяющий разработку на соответствие установленным требованиям;

- специалист кафедры, ответственный за проверку соответствия разработанного программного продукта, требованиям кафедры;

Полученные результаты – в результаты данного этапа стоит отнести конечную, полнофункциональную версию программного продукта, позволяющего автоматизировать составление и формирование технического задания.

4-й этап Тестирование программного продукта

На этапе тестирования производится тестирование программного продукта, разработанного в рамках мероприятий, сделанных на предыдущих этапах. Здесь речь идет уже об оценке функциональных возможностях программы, выявлении особенностей ее работы, выявлении сбоев и ошибок. На данном этапе программа должна тестироваться в режиме максимальной нагрузки и использованием всех ее функций.

Ключевые участники данного этапа:

- студент, ответственный за разработку проекта, в чьи обязанности входит: поддержка работоспособности программы и решение проблем пользователей;

- прочие пользователи, которые будут работать с программой.

Каждый привлеченный пользователь должен использовать программу в режиме максимальной нагрузки, задействовать все функции системы. Ввиду

того, что с системой будут работать только студенты и преподаватели кафедры, поэтому на данном этапе необходимо задействовать только их.

Результаты данного этапа – список особенностей функционирования разработанного программного продукта, выявление его слабых мест и возможных ошибок в работе.

5-й этап Эксплуатация и сопровождение программного продукта

На этапе эксплуатации программного продукта уже не идет речи о создании, внедрении и его разработке, здесь речь идет только о рабочих моментах, с которыми может столкнуться пользователь. На данном этапе работа пользователя с программой должна производиться в обычном, повседневном режиме, без принудительного увеличения нагрузки. Если, на данном этапе жизненного цикла у пользователя возникнут проблемы, со своими замечаниями, ему придется обратиться к инженерно-техническому персоналу.

Ключевые участники данного этапа:

- студент, в чьи обязанности входит: поддержка работоспособности программного продукта и решение проблем пользователей;
- прочие студенты кафедры, которые будут использовать программный продукт. Каждый пользователь должен использовать программу в режиме повседневной нагрузки.

Результаты данного этапа – список особенностей функционирования внедренной программы, ее слабые места, возможные ошибки в работе, а также список предпочтений, которые должны быть учтены при выходе новых обновлений.

6-й этап Развитие программного продукта

На данном этапе в разработанную и внедренную программу вносятся существенные и несущественные изменения, которые предполагают дополнение программы новыми функциями, увеличение ее производительности, изменение интерфейса. На данном этапе важен учет тех замечаний, что предоставляют пользователи программы в процессе ее эксплуатации. Сведения, полученные от пользователей, передаются разработчику программного продукта.

Ключевые участники данного этапа:

- студент, ответственный за разработку проекта, на данном этапе его обязанность – это корректировка разработанного плана, в связи с изменившимися условиями, а также устранение замечаний и возникающих в работе программы;
- научный руководитель, утверждающий необходимость внесения исправлений в программный продукт.

Результаты данного этапа – обновление функционала программы, ее интерфейса и возможностей. Также результатом является устранение старых проблем и ошибок в функционировании, но, при этом, возможно появление новых.

Таким образом выглядит модель разработки программного продукта, необходимого для автоматизации составления и формирования технического задания, которое осуществляется в рамках настоящего дипломного проекта. Более детально о разработке программы и особенностях ее эксплуатации будет сказано далее.

2.2 Разработка «Программы автоматизации процесса составления и формирования технического задания для выпускной квалификационной работы»

Разработка программного продукта осуществляется в 3 этапа.

На первом этапе разработки определяется содержание шаблона (см.рис.9 (а – г)), построение структуры, выделение перечня переменных и их типов (см.табл.5), определение используемых форм.

Таблица 5.
Перечень переменных

Переменная	Назначение	Тип
&Data i	Дата	Date
&cell i	Характеристика персонала	String
&tem1	Продукт разработки	String
&tem2	Требования к продукту	String

Переменная	Назначение	Тип
&tem3	Название системы	String
&tem4	Название организации-заказчика	String
&tem5	Назначение продукта	String
&tem6	Процессы и структуры, в которых предполагается использование продукта разработки	String
&tem7	Порядок взаимодействия с другими системами, возможности обмена информации	String
&tem8	Меры защиты информации	String
&tem9	Указание системного программного обеспечения	String
&pole1	Указание программного обеспечения, используемого для реализации	String
&pole2	Для сетевых систем – особенности реализации серверной и клиентской частей	String
&pole3	Форматы входных и выходных данных	String
&pole4	Источники данных и порядок их ввода в систему (программу), порядок вывода, хранения	String
&pole5	Общая характеристика пользовательского интерфейса	String
&pole6	Размещение информации на экране, дизайн экрана	String
&pole7	Особенности ввода информации пользователем, предоставление выходных данных	String
&pole8	Перечень сопроводительной документации	String
&pole9	Требования к содержанию отдельных документов	String
&img	Картинка	Image
&box i	Аппаратные требования	String

На втором этапе разработки необходимо создать шаблоны документов, на третьем этапе производится написание кода программы. Для работы программного продукта потребуются шаблоны в виде документов, созданных в программах MS Excel и MS Word.

Для создания шаблона в MS Word, необходимо создать новый текстовый документ в формате doc. Структура и содержание данного файла представлены на рисунке 9 (а – г).

1.1 Техническое задание
на разработку &tem1

«&tem3»

Составлен на основе ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы»

1. Общие сведения.

1.1. *Название организации-заказчика.*
&tem4

1.2. *Назначение продукта*
&tem5

1.3. *Плановые сроки начала и окончания работ.*

Таблица 1.
Календарный план

№ п/п	Этапы выполнения	Сроки выполнения этапа	
		Начало периода	Конец периода
1.	Разработка требований	&Data1	&Data2
2.	Разработка требований	&Data3	&Data4
3.	Реализация	&Data5	&Data6
4.	Тестирование	&Data7	&Data8
5.	Ввод в действие	&Data9	&Data0

2. Характеристика области применения продукта.

2.1. *Процессы и структуры, в которых предполагается использование продукта разработки*
Использование продукта предполагается в &tem6.

2.2. *Характеристика персонала (количество, квалификация, степень готовности)*

а)

	Квалификация	Степень готовности
Рабочие	&Cell1	&Cell2
Руководители	&Cell3	&Cell4
Специалисты	&Cell5	&Cell6
Технические исполнители	&Cell7	&Cell8

- 3. Требования к продукту разработки.**
- 3.1. *Требования к продукту в целом.*
&tem2.
- 3.2. *Аппаратные требования.*

Таблица 3.
Аппаратные требования

	Минимальные	Рекомендуемые
Процессор	&Box1	&Box2
Оперативная память	&Box3	&Box4
Видеоадаптер и монитор	&Box5	&Box6
Свободное место на HDD	&Box7	&Box3

- 3.3. *Указание системного программного обеспечения*
&tem9.
- 3.4. *Указание программного обеспечения, используемого для реализации.*
&pole1.
- 3.5. *Для сетевых систем – особенности реализации серверной и клиентской частей.*
&pole2.
- 3.6. *Форматы входных и выходных данных.*
&pole3.

б)

- 3.7. Источники данных и порядок их ввода в систему (программу),
порядок вывода, хранения
&role4.
- 3.8. Порядок взаимодействия с другими системами, возможности
обмена информацией.
&tem7.
- 3.9. Меры защиты информации:
&tem8
4. Требования к пользовательскому интерфейсу.
- 4.1. Общая характеристика пользовательского интерфейса.
&role5.
- 4.2. Размещение информации на экране, дизайн экрана.
&role6.
- &img
5. Требования к документированию.
- 5.1. Перечень сопроводительной документации.
&role8.
- 5.2. Требования к содержанию отдельных документов.
&role9.
6. Порядок сдачи-приемки продукта.

Рисунок 1. &img

- 4.3. Особенности ввода информации пользователем, предоставление
выходных

в)

г)

Рисунок 9. Внешний вид шаблона MS Word

Данный шаблон оформлен в соответствии с требованиями стандарта на оформление технического задания и используется для заполнения данными из программы, с помощью которой осуществляется автоматизированное формирование ТЗ.

Второй шаблон, который требуется для работы программы – это шаблон, сформированный в программе MS Excel. Данный шаблон используется для ввода исходных, обрабатываемых программой. Внешний вид данного шаблона представлен на рисунке 10.

Техническое задание

на разработку поле - ComboBox

Название системы: поле - ячейка, тип - string

1. Общие сведения

1) Название организации-заказчика поле - ячейка, тип - string

2) Назначение системы поле - ячейка, тип - string

3) Плановые сроки начала и окончания работ

Этап выполнения	Начало периода	Конец периода
1. Разработка требований	&Data1	&Data2
2. Проектирование	&Data3	&Data4
3. Реализация	&Data5	&Data6
4. Тестирование	&Data7	&Data8
5. Ввод в действие	&Data9	&Data0

} поле - ячейка, тип - date

2. Характеристика области применения продукта

1) Процессы и структуры, в которых предполагается использование продукта разработки

Использование продукта предполагается в поле - ячейка, тип - string

2) Характеристика персонала

поле - CheckBox

	Квалификация	Степень готовности
<input type="checkbox"/> Рабочие	&Cell11	&Cell12
<input type="checkbox"/> Руководители	&Cell13	&Cell14
<input type="checkbox"/> Специалисты	&Cell15	&Cell16
<input type="checkbox"/> Технические исполнители	&Cell17	&Cell18

Рисунок 10. Поля для ввода данных

Как можно видеть оба представленных шаблона имеют одинаковые поля, которые содержат отдельные строки технического задания, формируемого для разработки программных продуктов. Но отличие шаблона в файле, подготовленном в программе MS Excel, заключается в том, что данный файл также будет содержать исполняемый код программы, разработанный с помощью встроенного языка программирования VBA.

Перед началом разработки программы, необходимо добавить вкладку «Разработчик» в область вкладок MS Excel. Для этого необходимо выполнить следующие действия²:

² [https://msdn.microsoft.com/ru-ru/library/office/ee814737\(v=office.14\).aspx](https://msdn.microsoft.com/ru-ru/library/office/ee814737(v=office.14).aspx)

- 1) На вкладке Файл выберите Параметры, чтобы открыть диалоговое окно Параметры Excel.
- 2) Щелкните Настройка ленты в левой части диалогового окна.
- 3) В разделе Выбрать команды из, расположенном слева в окне, выберите Популярные команды.
- 4) В разделе Настройка ленты, который находится справа в диалоговом окне, выберите Основные вкладки, а затем установите флажок Разработчик (см.рис.11).
- 5) Нажмите кнопку ОК.

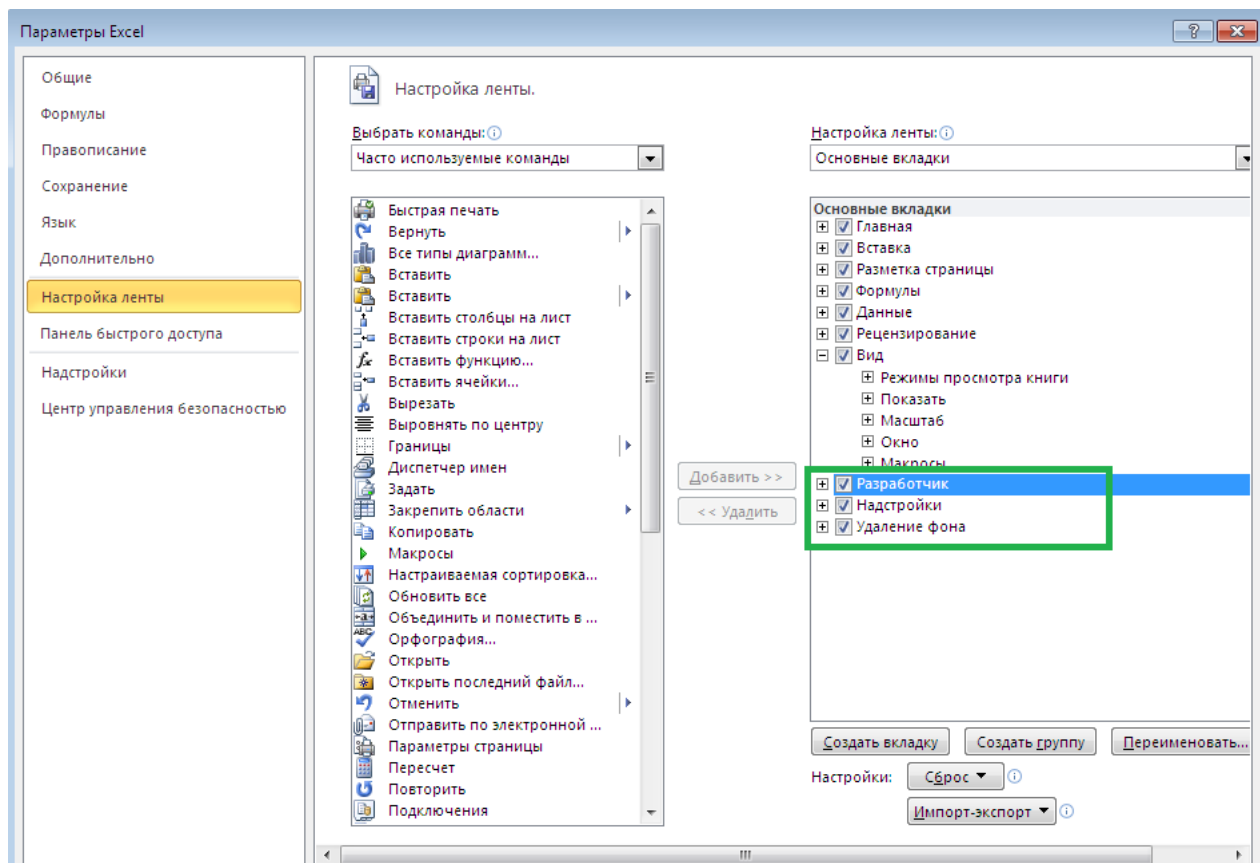


Рисунок 11. Включение вкладки «Разработчик»

После проделанных действий среди всех существующих вкладок, должна появиться новая вкладка «Разработчик», внешний вид которой представлен на рисунке 12.

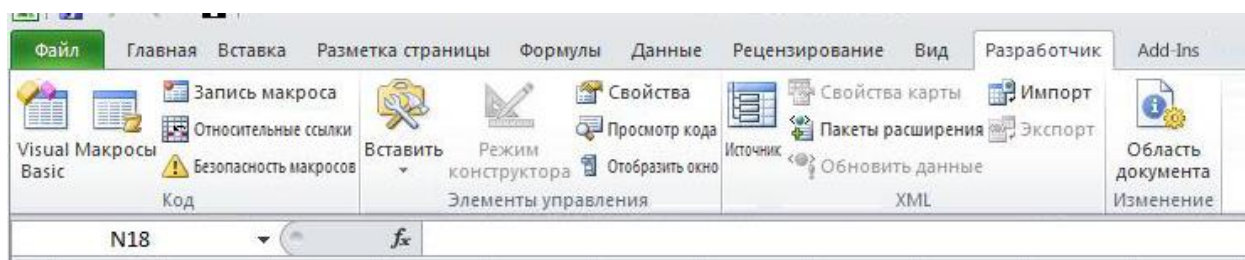


Рисунок 12. Внешний вид вкладки «Разработчик»

После того, как функционал разработчика стал доступен можно переходить к непосредственному написанию исходного кода программы. Для этого необходимо на вкладке «Разработчик» найти кнопку «Visual Basic», и нажать ее (см.рис.12). После открытия редактора программного кода, необходимо начать редактирование кода для объекта «Лист1», т.е. для текущего excel-файла. Для написания кода двойным щелчком левой кнопки мыши нажать на необходимый объект, в результате чего откроется меню для ввода программного кода, как показано на рисунке 13.

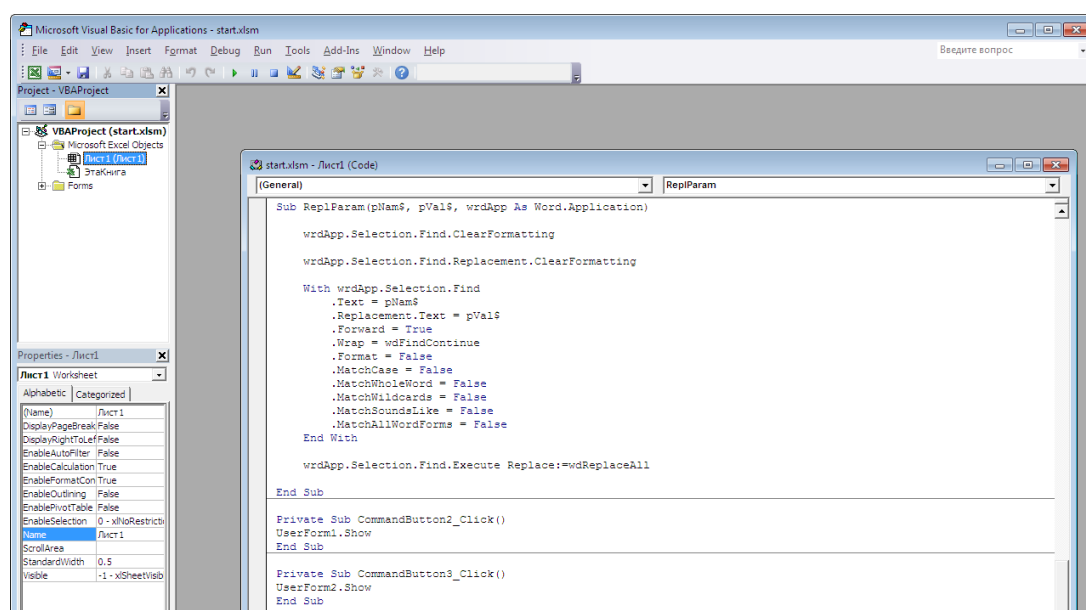


Рисунок 13. Редактор программного кода VBA

После заполнения полей информацией в шаблоне MS Excel и нажатие кнопки ОК, вызывается процедура обработки события, в которой переменным присваиваются поля с информацией.

Ниже представлен краткий листинг присвоения переменным полей с информацией.

```
tem1$ = ComboBox1.Value  
tem3$ = Cells(24, 53).Value  
tem4$ = Cells(46, 53).Value  
tem5$ = Cells(57, 53).Value
```

После этого в процедуру ReplParam передаются значения полей, в которой происходит поиск и замена полей в шаблоне MS Word.

Ниже представлен краткий листинг процедуры ReplParam.

```
Sub ReplParam(pNam$, pVal$, wrdApp As Word.Application)  
    wrdApp.Selection.Find.ClearFormatting  
    wrdApp.Selection.Find.Replacement.ClearFormatting  
    With wrdApp.Selection.Find  
        .Text = pNam$  
        .Replacement.Text = pVal$  
        .Forward = True  
        .Wrap = wdFindContinue  
        .Format = False  
        .MatchCase = False  
        .MatchWholeWord = False  
        .MatchWildcards = False  
        .MatchSoundsLike = False  
        .MatchAllWordForms = False  
    End With  
    wrdApp.Selection.Find.Execute Replace:=wdReplaceAll  
End Sub
```

После написания исходного кода программы (см. листинг 1 – 9) необходимо подключить библиотеки, которые требуются для работы разработанного программного продукта. Для этого необходимо войти в меню Tools главного меню редактора и выбрать пункт Refences, где необходимо подключить библиотеки согласно рисунку 14.

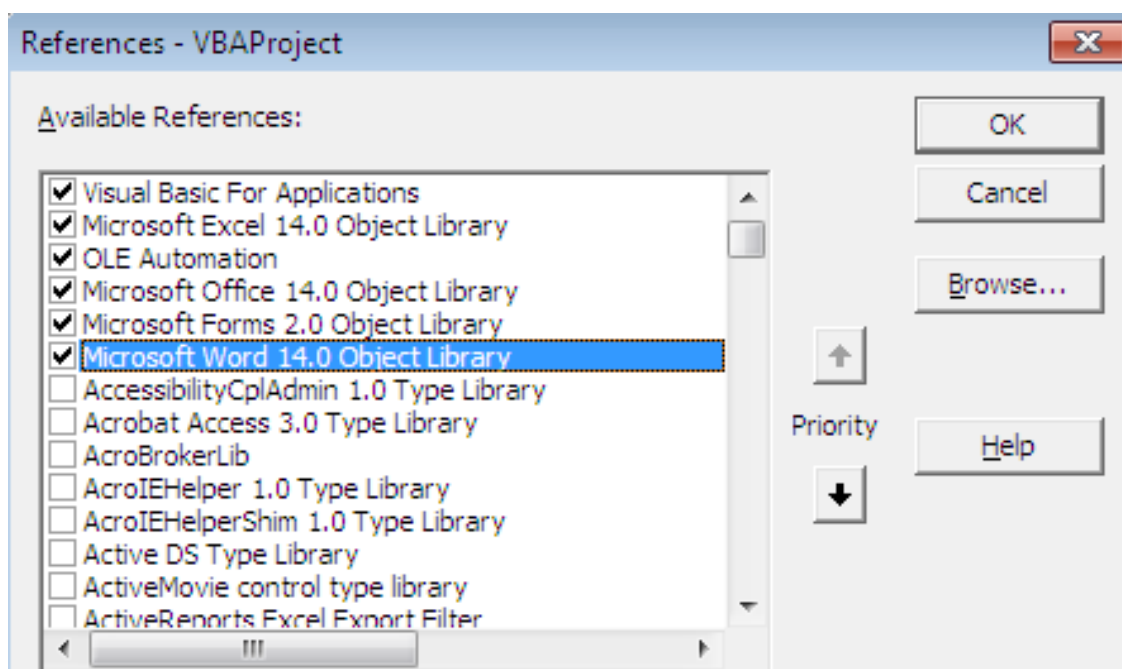


Рисунок 14. Подключение дополнительных библиотек

После того, как программный код полностью написан (см. листинг 1 – 9), необходимо его протестировать. Для этого необходим пробный запуск программы. В ряде случаев возможно возникновение ошибок, о которых необходимо сообщить разработчику. Непосредственное тестирование полученного программного продукта и составление документации к нему рассмотрим в следующем разделе дипломного проекта.

2.3 Техническая документация

Рассмотрим руководство пользователя программы.

Содержание

1.	ЗАПУСК ПРОГРАММЫ.....	1
2.	ВВОД ДАННЫХ.....	1
3.	ФОРМИРОВАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ.....	2
4.	ЗАПУСК ГОТОВОГО ФАЙЛА	2
5.	ВОЗМОЖНЫЕ ОШИБКИ.....	3

Рисунок 1. Содержание технической документации программы «Автоматизация процесса составления и формирования технического задания для выпускной квалификационной работы»

1. Запуск программы

Для запуска программы необходимо открыть файл start.xlsm

2. Ввод данных

Для ввода исходных данных, на основе которых будет формироваться техническое задание, необходимо заполнить поля, указанные в файле. Для заполнения полей необходимо использовать компьютерную клавиатуру и манипулятор мыш. Представленные для заполнения поля делятся на несколько типов:

- 1) Поля для ввода данных, представлены на рисунке 2

Название системы:	ИНФОРМАЦИОННАЯ СИСТЕМА
1. Общие сведения	
1) Название организации-заказчика	ООО "Компания"
2) Назначение системы	3

Рисунок 2. Поля для ввода данных

- 2) Поля для выбора данных, представлены на рисунке 3.

8) Порядок взаимодействия с другими системами, возможности обмена информации	<input checked="" type="radio"/> Имеется <input type="radio"/> Не имеется	36
9) Меры защиты информации	<input checked="" type="checkbox"/> Ограничение прав доступа <input checked="" type="checkbox"/> Аутентификация <input type="checkbox"/> Защита данных паролем	

Рисунок 3. Поля для выбора данных

3. Формирование технического задания

После того как все поля технического задания заполнены, необходимо сформировать конечный файл, для этого необходимо нажать кнопку «ОК», в

появившемся окне (см.рис.4) ввести название будущего файла и нажать кнопку «ОК».

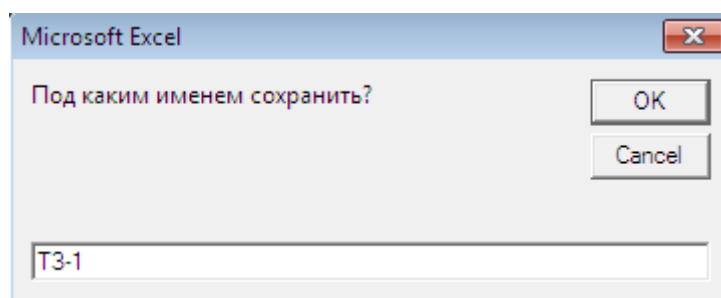


Рисунок 4. Ввод имени файла

4. Запуск готового файла

В папке с программой создается файл, в формате doc, с названием ТЗ-1, который можно открыть для сверки его с введенными данными. На рисунке 5 представлено содержание данного файла.

1.1 Техническое задание на разработку Информационной системы

«1».

Составлен на основе ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы»

1. Общие сведения.

1.1. Название организации-заказчика.

ООО «Заря»

1.2. Назначение ИС

3

1.3. Плановые сроки начала и окончания работ.

Таблица 1.
Календарный план

№ п/п	Этапы выполнения	Сроки выполнения этапа	
		<u>Начало</u> периода	<u>Конец</u> периода
1.	Разработка требований	25.01.2016	26.01.2016

Рисунок 5. Пример сформированного ТЗ

На представленном рисунке видно, что заполнены только 2 поля – пункт 1.1 и пункт 1 таблицы 1. Это сделано только для того, чтобы проверить работоспособность разработанного программного продукта.

5. Возможные ошибки

При работе разработанного продукта возможно возникновение ошибки, представленной на рисунке 6.

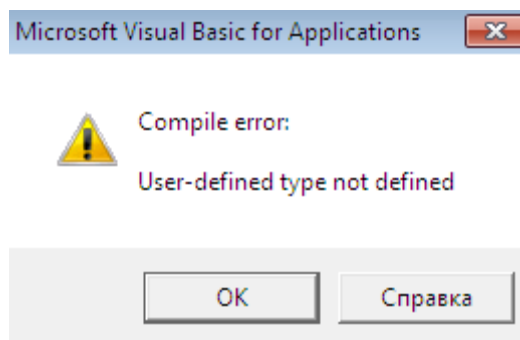


Рисунок 6. Ошибка при работе программы

Возникновение данной ошибки связано с тем, что в программном продукте отключена библиотека Microsoft Word X.X Object Library (где X.X – версия библиотеки в зависимости от версии MS Office). Для подключения данной библиотеки, необходимо кнопкой Stop (■) приостановить работу программы, зайти в меню Tool->References и включить поддержку данной библиотеки.

Подводя итоги работы, проделанной в рамках данной главы дипломного проекта, можно сделать вывод о том, что программный продукт, который необходимо разработать был формализован в виде IDEF0-модели, был представлен алгоритм его работы. Также были рассмотрены основные этапы жизненного цикла программного продукта и мероприятия, выполняемые на каждом из этапов. По завершению разработки было создано руководство пользователя. Таким образом, можно говорить об успешном завершении разработки программного продукта.

2.4 Апробация результатов работы

Материалы работы прошли апробацию и были представлены в виде публикации в межвузовском сборнике научных работ.

Приложение 1. Голяка М. А., Сардак Л.В. Автоматизация процесса составления и формирования технического задания студентами инженерных направлений подготовки // Актуальные вопросы преподавания математики, информатики и информационных технологий: межвузовский сборник научных работ. Екатеринбург: Урал. гос. пед. ун-т, 2016. - С. 23-30.

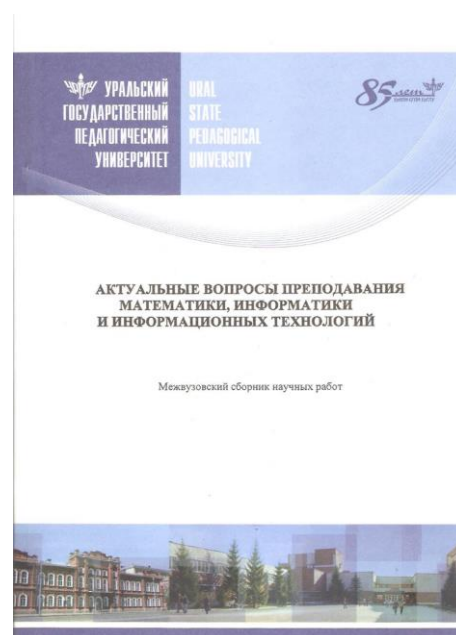


Рисунок 15. Обложка сборника «Актуальные вопросы преподавания математики, информатики и информационных технологий»

Заключение

В процессе выполнения ВКР проделана работа по изучению основных аспектов такой области знаний как «разработка программного обеспечения и информационных систем». Особое внимание было уделено формализации данного процесса, а именно – составлению технического задания. На основании проведенного исследования было проведено моделирование предметной области, с последующей разработкой программного продукта, который позволяет формировать программную документацию в автоматическом режиме, на основе параметров, задаваемых пользователем программы.

Таким образом, можно говорить о том, что основная цель дипломного проектирования – разработать программу для автоматизации процесса составления и формирования технического задания для выпускных квалификационных работ была достигнута в полной мере.

Для достижения поставленной цели в рамках проделанной работы были решены следующие задачи:

- проведен анализ предметной области;
- проведен анализ документации, формализующей предметную область;
- разработан программный продукт и документация на него;
- проведена апробация программного продукта.

Полученный программный продукт полностью соответствует техническому заданию, которое было разработано специально для его создания.

Выполненная работа, как и было, сказано ранее является актуальной, т.к. в ней рассматриваются актуальные вопросы, касающиеся реальной практики разработки программного обеспечения. Дальнейшее развитие работы возможно, за счет добавления новых функций в программный продукт, его развитие, в более сложную систему, с возможностью хранения библиотеки шаблонов технических заданий для более сложных программных систем, либо за счет внедрения функций управления проектом разработки программного обеспечения. Также возможно внедрение и иных функций, но на текущий момент

работа является полностью завершенной, соответствует всем заданным требованиям и способна выполнять весь возложенный на нее функционал.

Список информационных источников

1. Antiloop, Что такое DirectX? [Электронный ресурс] URL: [http://cit-forum.web.am/programming /digest/directxwhatis.shtml](http://cit-forum.web.am/programming/digest/directxwhatis.shtml) (дата обращения 05.03.2016)
2. Borland Delphi 8. [Электронный ресурс] URL: <http://www.interface.ru/borland/delphi8.htm> (дата обращения 08.03.2016)
3. Аветисов, М.А. Электронная доставка документов - создание архива сканированных материалов с поисковым аппаратом // Науч. и техн. Б-ки. - 2010. - № 3. - с. 7-11.
4. Андерсон К. Локальные сети. Полное руководство / К. Андерсон, Минаси М - СПб.: КОРОНА принт, 2009. - 624 с.
5. Артемов Д. Моделирование успешной компании. [Электронный ресурс] URL: <http://www.cia-center.ru/cc/portal/ss/ss05.html> (дата обращения 03.03.2016)
6. Афанасьев А.А. Аутентификация. Теория и практика обеспечения безопасного доступа к информационным ресурсам, Горячая Линия – Телеком, 2012 – 550с.
7. Басовский Е.Л., Протасьев В.Б. Управление качеством. Учебник. – М.: «Инфра». «Высшее образование», 2013. – 214 с.
8. Вежнев В. Краткое неформальное введение в графику Windows. [Электронный ресурс] URL: <http://graphics.cs.msu.su/courses/cg/library/windows/> (дата обращения 08.03.2016)
9. Гарнаев А. Самоучитель VBA. – СПб.: БХВ-Петербург, 2004. – 244 с.
10. Голдовский А., Голдовский П., Федотов О. Система DELMIA как решение для моделирования производственной деятельности. [Электронный ресурс] URL: http://www.cadcamcae.lv/hot/delmia_n19_p13.htm (дата обращения 05.03.2016)

11. Ехлаков Ю.П. Вывод прикладного программного обеспечения на рынке корпоративных продаж: взгляд разработчика // «Маркетинг в России и за рубежом» - 2014. - №4. с. 20-23
12. Жизненный цикл информационных систем. [Электронный ресурс] URL: <http://www.abn.ru/inf/setevoi/cycle.shtml> (дата обращения 05.03.2016)
13. К. Ауэр, Р. Миллер. Экстремальное программирование: постановка процесса с первых шагов и до победного конца. СПб.: Питер, 2013 - 715 с.
14. Кинг Н. Не просто новая версия — иное видение. [Электронный ресурс] URL: <http://www.interface.ru/fset.asp?Url=/microsoft/news/m030228139.htm> (дата обращения 13.03.2016)
15. Кобёрн А. Современные методы описания функциональных требований к системам. — М.: Лори, 2013. - 510 с.
16. Леффингуелл Д., Уидриг Д. Принципы работы с требованиями к программному обеспечению. — М.: Вильямс, 2012. - 414 с.
17. Липаев, В.В. Программная инженерия. Методологические основы: Учеб. / В. В. Липаев; Гос. ун-т — Высшая школа экономики. — М.: ТЕИС, 2011. — 608 с.
18. Лифиц И.М. Основы стандартизации метрологии и сертификации. Учебник. — М.: «Юрайт», 2013. — 288 с.
19. М.Кон. Scrum: гибкая разработка ПО. - М.:«Вильямс», 2011. - 576 с.
20. Мухина М.К. Изучение стиля жизни потребителей и сегментирование рынка на основе психографических типов // Маркетинг в России и за рубежом. 2014. № 3(17). - с. 3-20.
21. Орлик С. "Основы программной инженерии" на базе IEEE Guide to SWEBOOK [Электронный ресурс] URL: <http://swebok.sorlik.ru> (дата обращения 03.03.2016)

22. Особенности процессов жизненного цикла программных средств в стандарте ISO 12207. [Электронный ресурс] URL: <http://www.jetinfo.ru/2004/3/1/article1.3.2004504.html> (дата обращения 13.03.2016)
23. Преимущества Borland C++ Builder перед другими средствами разработки. [Электронный ресурс] URL: <http://emanual.ru/cgi-bin/get.pl?id=5178&format=show> (дата обращения 03.03.2016)
24. Прикладное программное обеспечение [Электронный ресурс] URL: https://ru.wikipedia.org/wiki/Прикладное_программное_обеспечение (дата обращения 02.02.2016)
25. Процесс разработки программного обеспечения [Электронный ресурс] URL: <http://www.intuit.ru/studies/courses/497/353/lecture/8406> (дата обращения 13.03.2016)
26. Репин В. Бизнес-процессы: моделирование, внедрение, управление. – М.: Манн, Иванов, Фербер, 2012. – 512 с.
27. С. Макконнелл. Влияние итеративных подходов на предварительные условия // Совершенный код = Code Complete. – СПб.: Русская Редакция, 2014. — с. 31. — 896 с.
28. Сайт компании "СОФТПРОМ". [Электронный ресурс] URL: http://www.softprom.com/products_detail.phtml?products_id=30&category_id=8
29. Хореев В.Д. Самоучитель программирования на VBA в Microsoft Office. – М.: Юниор, 2001. – 341 с.
30. Чейз Р.Б. Производственный и операционный менеджмент. М.: Издательское объединение «ЮНИТИ», 2004. – 410 с.
31. Шеер А.-В. ARIS - моделирование бизнес-процессов. - М.: Вильямс, 2009. - 224 с.
32. Щербаченко М.Г. Использование имитационного моделирования в процессе составления бюджета промышленного предприятия. [Электронный ресурс] URL: http://www.cis2000.ru/publish/articles/book_33/intro.shtml (дата обращения 03.02.2016)

Приложения

Приложение 1.

Техническая документация к программе «Автоматизация процесса составления и формирования технического задания для выпускной квалификационной работы»

Содержание

1. ЗАПУСК ПРОГРАММЫ.....	1
2. ВВОД ДАННЫХ.....	1
3. ФОРМИРОВАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ.....	2
4. ЗАПУСК ГОТОВОГО ФАЙЛА.....	2
5. ВОЗМОЖНЫЕ ОШИБКИ.....	3

Рисунок 1. Содержание технической документации программы «Автоматизация процесса составления и формирования технического задания для выпускной квалификационной работы»

1. Запуск программы

Для запуска программы необходимо открыть файл start.xlsm

2. Ввод данных

Для ввода исходных данных, на основе которых будет формироваться техническое задание, необходимо заполнить поля, указанные в файле. Для заполнения полей необходимо использовать компьютерную клавиатуру и манипулятор мыш. Представленные для заполнения поля делятся на несколько типов:

2) Поля для ввода данных, представлены на рисунке 2

Название системы:

ИНФОРМАЦИОННАЯ СИСТЕМА

1. Общие сведения

1) Название организации-заказчика

ООО "КомпаниЯ"

2) Назначение системы

3

Рисунок 2. Поля для ввода данных

2) Поля для выбора данных, представлены на рисунке 3.

8) Порядок взаимодействия с другими системами, возможности обмена информации	<input checked="" type="radio"/> Имеется <input type="radio"/> Не имеется	36
9) Меры защиты информации	<input checked="" type="checkbox"/> Ограничение прав доступа <input checked="" type="checkbox"/> Аутентификация <input type="checkbox"/> Защита данных паролем	

Рисунок 3. Поля для выбора данных

3. Формирование технического задания

После того как все поля технического задания заполнены, необходимо сформировать конечный файл, для этого необходимо нажать кнопку «ОК», в появившемся окне (см.рис.4) ввести название будущего файла и нажать кнопку «ОК».

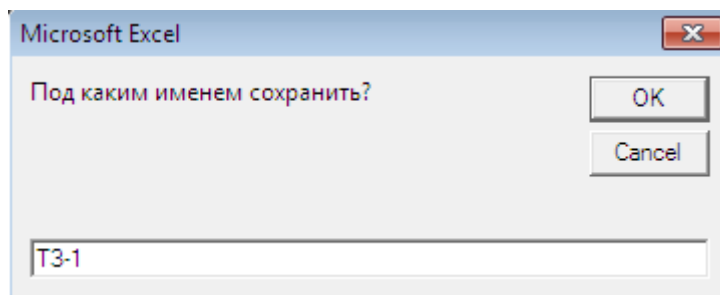


Рисунок 4. Ввод имени файла

4. Запуск готового файла

В папке с программой создается файл, в формате doc, с названием ТЗ-1, который можно открыть для сверки его с введенными данными. На рисунке 5 представлено содержание данного файла.

1.1 Техническое задание на разработку Информационной системы

«1».

Составлен на основе ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы»

1. Общие сведения.

1.1. Название организации-заказчика.
ООО «Заря»

1.2. Назначение ИС

3

1.3. Плановые сроки начала и окончания работ.

Таблица 1.
Календарный план

№ п/п	Этапы выполнения	Сроки выполнения этапа	
		Начало периода	Конец периода
1.	Разработка требований	25.01.2016	26.01.2016

Рисунок 5. Пример сформированного ТЗ

На представленном рисунке видно, что заполнены только 2 поля – пункт 1.1 и пункт 1 таблицы 1. Это сделано только для того, чтобы проверить работоспособность разработанного программного продукта.

5. Возможные ошибки

При работе разработанного продукта возможно возникновение ошибки, представленной на рисунке 6.

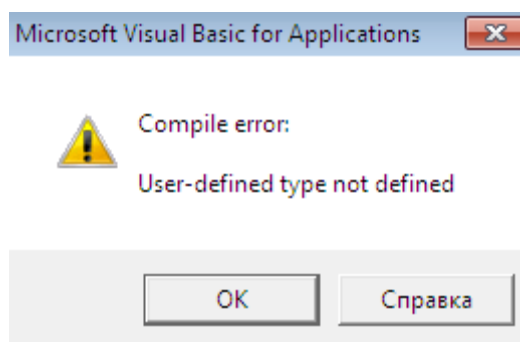



Рисунок 6. Ошибка при работе программы

Возникновение данной ошибки связано с тем, что в программном продукте отключена библиотека Microsoft Word X.X Object Library (где X.X – версия библиотеки в зависимости от версии MS Office). Для подключения данной библиотеки, необходимо кнопкой Stop () приостановить работу программы, зайти в меню Tool->References и включить поддержку данной библиотеки.

Приложение 2.

Листинг 1.

Процедура обработки события CommandButton1_Click()

```
Private Sub CommandButton1_Click()  
Dim wrdApp As Word.Application 'экземпляр приложения  
Dim wrdDoc As Word.Document 'экземпляр документа  
Dim wrdFind As Word.Find 'Объект Find предназначен для выполнения операции  
поиска  
  
    'вычисление пути расположения файла шаблона MS Word  
    HomeDir$ = ThisWorkbook.Path  
    Set wrdApp = CreateObject("Word.Application")  
    Set wrdDoc = wrdApp.Documents.Open(HomeDir$ & "\pattern.docx")  
    'присвоение переменным значения ячеек  
    tem1$ = ComboBox1.Value  
    tem3$ = Cells(24, 53).Value  
    tem4$ = Cells(46, 53).Value  
    tem5$ = Cells(57, 53).Value  
    Data1$ = Cells(74, 105).Value + 1  
    Data2$ = Cells(74, 131).Value + 1  
    Data3$ = Cells(80, 105).Value + 1  
    Data4$ = Cells(80, 131).Value + 1  
    Data5$ = Cells(86, 105).Value + 1  
    Data6$ = Cells(86, 131).Value + 1  
    Data7$ = Cells(92, 105).Value + 1  
    Data8$ = Cells(92, 131).Value + 1  
    Data9$ = Cells(98, 105).Value + 1  
    Data0$ = Cells(98, 131).Value + 1  
  
    tem6$ = Cells(131, 53).Value  
    If (OptionButton1.Value = True) Then  
        tem7$ = Cells(308, 79).Value  
    End If  
    If (OptionButton2.Value = True) Then  
        tem7$ = "Система не осуществляет взаимодействие с другими системами"  
    End If  
    If (CheckBox1.Value = True) Then  
        tem8$ = tem8$ + CheckBox1.Caption & vbCrLf  
    End If  
    If (CheckBox2.Value = True) Then  
        tem8$ = tem8$ + CheckBox2.Caption & vbCrLf  
    End If  
    If (CheckBox3.Value = True) Then
```

```

tem8$ = tem8$ + CheckBox3.Caption & vbCrLf
End If
If (CheckBox4.Value = True) Then
cell11$ = Cells(157, 79).Value
cell12$ = Cells(157, 131).Value
Else
cell11$ = ""
cell12$ = ""
End If
If (CheckBox5.Value = True) Then
cell13$ = Cells(158, 79).Value
cell14$ = Cells(158, 131).Value
Else
cell13$ = ""
cell14$ = ""
End If
If (CheckBox6.Value = True) Then
cell15$ = Cells(159, 79).Value
cell16$ = Cells(159, 131).Value
Else
cell15$ = ""
cell16$ = ""
End If
If (CheckBox7.Value = True) Then
cell17$ = Cells(160, 79).Value
cell18$ = Cells(160, 131).Value
Else
cell17$ = ""
cell18$ = ""
End If
tem2$ = Cells(173, 53).Value
box1$ = Cells(190, 105).Value
box2$ = Cells(190, 131).Value
box3$ = Cells(191, 105).Value
box4$ = Cells(191, 131).Value
box5$ = Cells(192, 105).Value
box6$ = Cells(192, 131).Value
box7$ = Cells(193, 105).Value
box8$ = Cells(193, 131).Value
tem9$ = Cells(195, 53).Value
pole1$ = Cells(213, 53).Value
pole2$ = Cells(239, 53).Value

```

```

pole3$ = Cells(264, 53).Value
pole4$ = Cells(282, 53).Value
pole5$ = Cells(373, 53).Value
pole6$ = Cells(394, 53).Value
pole7$ = Cells(416, 53).Value
pole8$ = Cells(453, 53).Value
pole9$ = Cells(474, 53).Value
'передача процедуре ReplParam переменных
ReplParam "&tem1", tem1$, wrdApp
ReplParam "&tem3", tem3$, wrdApp
ReplParam "&tem4", tem4$, wrdApp
ReplParam "&tem5", tem5$, wrdApp
ReplParam "&Data1", Data1$, wrdApp
ReplParam "&Data2", Data2$, wrdApp
ReplParam "&Data3", Data3$, wrdApp
ReplParam "&Data4", Data4$, wrdApp
ReplParam "&Data5", Data5$, wrdApp
ReplParam "&Data6", Data6$, wrdApp
ReplParam "&Data7", Data7$, wrdApp
ReplParam "&Data8", Data8$, wrdApp
ReplParam "&Data9", Data9$, wrdApp
ReplParam "&Data0", Data0$, wrdApp
ReplParam "&tem6", tem6$, wrdApp
ReplParam "&tem7", tem7$, wrdApp
ReplParam "&tem8", tem8$, wrdApp
ReplParam "&tem2", tem2$, wrdApp
ReplParam "&tem9", tem9$, wrdApp
ReplParam "&pole1", pole1$, wrdApp
ReplParam "&pole2", pole2$, wrdApp
ReplParam "&pole3", pole3$, wrdApp
ReplParam "&pole4", pole4$, wrdApp
ReplParam "&pole5", pole5$, wrdApp
ReplParam "&pole6", pole6$, wrdApp
ReplParam "&pole7", pole7$, wrdApp
ReplParam "&pole8", pole8$, wrdApp
ReplParam "&pole9", pole9$, wrdApp
ReplParam "&cell1", cell1$, wrdApp
ReplParam "&cell2", cell2$, wrdApp
ReplParam "&cell3", cell3$, wrdApp
ReplParam "&cell4", cell4$, wrdApp
ReplParam "&cell5", cell5$, wrdApp
ReplParam "&cell6", cell6$, wrdApp

```

```

ReplParam "&cell7", cell7$, wrdApp
ReplParam "&cell8", cell8$, wrdApp
ReplParam "&box1", box1$, wrdApp
ReplParam "&box2", box2$, wrdApp
ReplParam "&box3", box3$, wrdApp
ReplParam "&box4", box4$, wrdApp
ReplParam "&box5", box5$, wrdApp
ReplParam "&box6", box6$, wrdApp
ReplParam "&box7", box7$, wrdApp
ReplParam "&box8", box8$, wrdApp
'сохранение документа
nameOut$ = InputBox("Под каким именем сохранить?")
wrdDoc.SaveAs HomeDir$ & "\" & nameOut$
wrdDoc.Close
wrdApp.Quit
Set wrdDoc = Nothing
Set wrdApp = Nothing
MsgBox "Ok!"
End Sub

'процедура поиска и замены в шаблоне MS Word
Sub ReplParam(pNam$, pVal$, wrdApp As Word.Application)
wrdApp.Selection.Find.ClearFormatting
wrdApp.Selection.Find.Replacement.ClearFormatting
With wrdApp.Selection.Find
.Text = pNam$
.Replacement.Text = pVal$
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
wrdApp.Selection.Find.Execute Replace:=wdReplaceAll
End Sub

```

Листинг 2.

Процедура обработки события CommandButton3_Click()

```

Private Sub CommandButton3_Click()
'Отображение объекта UserForm2
UserForm2.Show
End Sub

```


Процедура инициализации формы UserForm2

```
Private Sub UserForm_Initialize()  
    'заполнение списка ComboBox1 - ComboBox5 позициями используя метод  
    AddItem().  
    With UserForm2.ComboBox1  
        .Clear  
        .Value = ""  
        .AddItem "Название организации-заказчика"  
        .AddItem "Назначение системы"  
        .AddItem "Плановые сроки начала и окончания работ"  
    End With  
    With UserForm2.ComboBox2  
        .Clear  
        .Value = ""  
        .AddItem "Процессы и структуры, в которых предполагается использова-  
не продукта разработки"  
        .AddItem "Характеристика персонала"  
    End With  
    With UserForm2.ComboBox3  
        .Clear  
        .Value = ""  
        .AddItem "Требования к продукту в целом"  
        .AddItem "Аппаратные требования"  
        .AddItem "Указание системного программного обеспечения"  
        .AddItem "Указание программного обеспечения, используемого для ре-  
ализации"  
        .AddItem "Для сетевых систем - особенности реализации серверной и  
клиентской частей"  
        .AddItem "Форматы входных и выходных данных"  
        .AddItem "Источники данных и порядок их ввода в систему (программу),  
порядок вывода, хранения"  
        .AddItem "Порядок взаимодействия с другими системами, возможности об-  
мена информацией"  
        .AddItem "Меры защиты информации"  
    End With  
    With UserForm2.ComboBox4  
        .Clear  
        .Value = ""  
        .AddItem "Общая характеристика пользовательского интерфейса"  
        .AddItem "Размещение информации на экране, дизайн экрана"  
        .AddItem "Особенности ввода информации пользователем, предоставление  
выходных данных"
```

```

End With
With UserForm2.ComboBox5
    .Clear
    .Value = ""
    .AddItem "Перечень сопроводительной документации"
    .AddItem "Требования к содержанию отдельных документов"
End With
End Sub

```

Листинг 4.

Процедура обработки события ComboBox1_Change()

```

Private Sub ComboBox1_Change()
'загрузка в Image1 изображений, выбранных ComboBox1
HomeDir$ = ThisWorkbook.Path
If ComboBox1.Value = "Название организации-заказчика" Then
Image1.Picture = LoadPicture(HomeDir$ & "\Help\1.jpg")
ElseIf ComboBox1.Value = "Назначение системы" Then
Image1.Picture = LoadPicture(HomeDir$ & "\Help\2.jpg")
ElseIf ComboBox1.Value = "Плановые сроки начала и окончания работ" Then
Image1.Picture = LoadPicture(HomeDir$ & "\Help\3.jpg")
End If
End Sub

```

Листинг 5.

Процедура обработки события ComboBox2_Change()

```

Private Sub ComboBox2_Change()
'загрузка в Image2 изображений, выбранных ComboBox2
HomeDir$ = ThisWorkbook.Path
If ComboBox2.Value = "Процессы и структуры, в которых предполагается использование продукта разработки" Then
Image2.Picture = LoadPicture(HomeDir$ & "\Help\4.jpg")
ElseIf ComboBox2.Value = "Характеристика персонала" Then
Image2.Picture = LoadPicture(HomeDir$ & "\Help\5.jpg")
End If
End Sub

```

Листинг 6.

Процедура обработки события ComboBox3_Change()

```

Private Sub ComboBox3_Change()
'загрузка в Image3 изображений, выбранных ComboBox3
HomeDir$ = ThisWorkbook.Path
If ComboBox3.Value = "Требования к продукту в целом" Then
Image3.Picture = LoadPicture(HomeDir$ & "\Help\6.jpg")
ElseIf ComboBox3.Value = "Аппаратные требования" Then
Image3.Picture = LoadPicture(HomeDir$ & "\Help\7.jpg")

```

```

        ElseIf ComboBox3.Value = "Указание системного программного обеспечения"
Then
    Image3.Picture = LoadPicture(HomeDir$ & "\Help\8.jpg")
    ElseIf ComboBox3.Value = "Указание программного обеспечения, используе-
мого для реализации" Then
        Image3.Picture = LoadPicture(HomeDir$ & "\Help\9.jpg")
        ElseIf ComboBox3.Value = "Для сетевых систем - особенности реализации
серверной и клиентской частей" Then
            Image3.Picture = LoadPicture(HomeDir$ & "\Help\10.jpg")
            ElseIf ComboBox3.Value = "Форматы входных и выходных данных" Then
                Image3.Picture = LoadPicture(HomeDir$ & "\Help\11.jpg")
                ElseIf ComboBox3.Value = "Источники данных и порядок их ввода в систему
(программу), порядок вывода, хранения" Then
                    Image3.Picture = LoadPicture(HomeDir$ & "\Help\12.jpg")
                    ElseIf ComboBox3.Value = "Порядок взаимодействия с другими системами,
возможности обмена информацией" Then
                        Image3.Picture = LoadPicture(HomeDir$ & "\Help\13.jpg")
                        ElseIf ComboBox3.Value = "Меры защиты информации" Then
                            Image3.Picture = LoadPicture(HomeDir$ & "\Help\14.jpg")
                            End If
                        End Sub

```

Листинг 7.

Процедура обработки события ComboBox4_Change()

```

Private Sub ComboBox4_Change()
'загрузка в Image4 изображений, выбранных ComboBox4
HomeDir$ = ThisWorkbook.Path
If ComboBox4.Value = "Общая характеристика пользовательского интерфейса"
Then
    Image4.Picture = LoadPicture(HomeDir$ & "\Help\15.jpg")
    ElseIf ComboBox4.Value = "Размещение информации на экране, дизайн экрана"
Then
        Image4.Picture = LoadPicture(HomeDir$ & "\Help\16.jpg")
        ElseIf ComboBox4.Value = "Особенности ввода информации пользователем,
предоставление выходных данных" Then
            Image4.Picture = LoadPicture(HomeDir$ & "\Help\17.jpg")
            End If
        End Sub

```

Листинг 8.

Процедура обработки события ComboBox5_Change()

```

Private Sub ComboBox5_Change()
'загрузка в Image5 изображений, выбранных ComboBox5
HomeDir$ = ThisWorkbook.Path
If ComboBox5.Value = "Перечень сопроводительной документации" Then

```

```

Image5.Picture = LoadPicture(HomeDir$ & "\Help\18.jpg")
ElseIf ComboBox5.Value="Требования к содержанию отдельных документов"Then
Image5.Picture = LoadPicture(HomeDir$ & "\Help\19.jpg")
End If
End Sub

```

Листинг 9.

Процедура обработки события Workbook_Open()

```

Private Sub Workbook_Open()
'заполнение списка ComboBox1 позициями используя метод AddItem().
    With Лист1.ComboBox1
        .Clear
        .Value = ""
        .AddItem "операционной системы"
        .AddItem "сервисной программы"
        .AddItem "языка программирования"
        .AddItem "системы управления базами данных"
        .AddItem "средства для обработки цифровых данных"
        .AddItem "интегрированного пакета"
        .AddItem "игровой программы"
    End With
End Sub

```